RENARDUS: PROJECT DELIVERABLE

Project Number:	IST-1999-10562
Project Title:	Reynard - Academic Subject Gateway Service Europe
Deliverable Type:	Public

Deliverable Number:	D1.1
Contractual Date of Delivery:	31 March 2000
Actual Date of Delivery:	27 April 2000
Title of Deliverable:	Evaluation report of existing broker models in related projects
Workpackage contributing to the Deliverable:	WP1
Nature of the Deliverable:	Report
URL:	http://www.renardus.org/deliverables/
Authors:	Michael Day, Anders Ardö, Matthew J. Dovey, Martin Hamilton, Risto Heikkinen, Andy Powell and Arthur N. Olsen.
Contact Details:	Michael Day, UKOLN: the UK Office for Library and Information Networking, University of Bath, Bath BA2 7AY, UK. Email: m.day@ukoln.ac.uk, Phone: +44 1225 323923, Fax: +44 826838, URL: http://www.ukoln.ac.uk/

Abstract	Broker services enable the integration of distributed and heterogeneous information resources. The Renardus project will implement a Europe wide Internet information gateway service based on a generic broker architecture and data model that will allow the integrated searching and browsing of distributed resource collections. This report reviews eighteen broker architectures that have been developed for existing services and projects. An attempt has been made to map the function of each of these architectures (or broker models) onto the generic MODELS Information Architecture (MIA) and more specifically the MIA structure developed to describe services known as DNER Portals. The report concludes with some observations on the broker models reviewed and the protocols and software that they use.

Keywords	Broker models, Broker architectures, Renardus project, MODELS Information Architecture, MIA, Information gateways, Digital libraries

Distribution List:	Public
Issue:	1.0
Reference:	IST-1999-10562 / D1.1 / 1.0
Total Number of Pages:	90

TABLE OF CONTENTS

PART I TITLE PAGE

PART II - MANAGEMENT OVERVIEW

DOCUMENT CONTROL

Issue	*Date of Issue*	*Comments*
0.1	13 April 2000	First draft, for initial review by contributors.
0.2	17 April 2000	Second draft, for review by project partners.
1.0	27 April 2000	First public release.

EXECUTIVE SUMMARY

The object of the Renardus project is to establish an academic subject gateway service in Europe. The pilot system will be based on a generic broker-architecture and data-model that will allow the integrated searching and browsing of distributed resource collections.

For Renardus, it is important to ensure that any chosen solution is based on emerging developments rather than being constrained by decisions made by the subset of gateways that are participating in the initial stages of the project. This report, therefore, reviews eighteen broker models that have been developed for a variety of existing services, projects and initiatives. The models were chosen because they were perceived to be relevant to the digital library context of Renardus.

Each broker model is briefly introduced and an attempt made to map its functions onto the generic model known as the MODELS Information Architecture (MIA). The MIA logical architecture is a layered architecture with five layers:

- Presenter
- Coordinator
- Mediator
- Communicator
- Provider

The functions of each model is analysed in relation to this logical architecture and notes made about the use of standards, protocols and software.

Eighteen broker models are reviewed:

- Agora
- Aquarelle
- Advanced Search Facility (ASF)
- CHIC-Pilot
- CORC - OCLC's Cooperative Online Resource Catalog
- DEF - Denmark's Electronic Research Library
- Die Digitale Bibliothek Nordrhein-Westfalen (NRW)
- ETB - the European Schools Treasury Broker
- EULER project
- Finnish Virtual Library (FVL)
- GAIA - Generic Architecture for Information Availability
- Harvest Indexer
- ht://Dig
- Isaac Network
- Jointly Administered Knowledge Environment (jake)
- Networked Computer Science Technical Research Library (NCSTRL)
- RDN ResourceFinder
- ROADS toolkit

- UNIverse

These can be arranged into the following main categories:

- The broker models that underlie open source indexing software toolkits like ASF Freeware, Harvest, ht://Dig, jake and ROADS.
- The broker models that underlie the cross searching of distributed Internet information gateways like the Finnish Virtual Library and the Resource Discovery Network (RDN) ResourceFinder. These currently tend to be based on open-source software like the ROADS toolkit and use relatively simple Internet protocols like WHOIS++ or LDAP.
- Broker models developed to handle more complex requirements, typically where more than one protocol and data format is in use. Some of those reviewed were based on - where possible - open source software, e.g. the broker developed for the EULER project. Some of the other systems are based to some extent on proprietary software and some have some dependence upon commercial products supplied by library software vendors. So, for example, the Agora Hybrid Library Management System (HLMS) is based on Fretwell-Downing Informatics's OLIB VDX system. CORC is based on proprietary software developed at OCLC, but could be licensed for use in a project like Renardus. These more complex models tend to be based on the action of standard protocols like Z39.50 and ISO ILL and sometimes need to interact with authentication services.
- A broker model being developed for 'information trading' (GAIA).

The review ends with some considerations towards determining an architectural model in Renardus and some conclusions on the broker review itself.

SCOPE STATEMENT

This report is the first public deliverable to be issued by WP1 (Functional Model) of the Renardus project. The objective of WP1 is to develop the architecture that will underpin the Renardus system. WP1, together with WP6 (Data Model and Data Flow) will provide the functional and data-model specifications of the Renardus broker system. To this end, WP1 has begun to analyse the functional requirements of the Renardus broker system from both service provider and end-user perspectives. These requirements were collected from a survey of Renardus participants and have been published as internal deliverable D1.2.

D1.1 should be able to provide background information for the development of a Renardus broker system based on the best current practice. Its findings will contribute to the specification of functional requirements for the Renardus system (internal deliverable D1.3) and ultimately, to the development of the architectural model for the Renardus system (internal deliverable D1.4 and public deliverable D1.5). D1.1 should also be able to provide background information for the development of the Renardus data model in WP6.

PART III - DELIVERABLE CONTENT

INTRODUCTION

This report will provide background information for the development of the Renardus architecture. It reviews 18 broker models that have been developed (or are being developed) for a variety of services, projects and other initiatives. Each broker model is introduced and then its functions are mapped to the generic broker architecture known as the MODELS Information Architecture (MIA) - and specifically to a diagram developed by Powell (1999) to provide an MIA view of a DNER Portal.

At the same time as this deliverable was being produced, Martin Hamilton of the Department of Computer Science at Loughborough University of Technology (LUT) was carrying out a similar review of broker models for a project funded by the NSF/JISC International Digital Libraries Initiative - the IMesh Toolkit project. Some of these reviews (ASF Freeware, CHIC-Pilot, Harvest, ht://Dig, Isaac Network, jake, ROADS) have - with kind permission - been adapted and included in this report. Other reviews have been provided by Anders Ardö of the Technical Knowledge Centre and Library of Denmark (DTV), Matthew J. Dovey of the University of Oxford Libraries Automation Service (LAS), Risto Heikkinen of Jyväskylä University Library, Andy Powell of UKOLN and by Arthur N. Olsen of NetLab. Production of the deliverable was co-ordinated by Michael Day of UKOLN.

Abbreviation	Organisation	Contributor/s
DTV	Technical Knowledge Centre and Library of Denmark, Lyngby, Denmark	Anders Ardö
LAS	University of Oxford Libraries Automation Service, Oxford, UK	Matthew J. Dovey
LUT	Department of Computer Science, Loughborough University, Loughborough, UK	Martin Hamilton
JyU	Jyväskylä University Library, Jyväskylä, Finland	Risto Heikkinen
NetLab	NetLab, Lund University Library, Lund, Sweden	Arthur N. Olsen
UKOLN	UKOLN: the UK Office for Library and Information Networking, University of Bath, Bath, UK	Michael Day, Andy Powell

Table: Summary of contributors and their organisational affiliations

GLOSSARY

ADAM
The Art, Design, Architecture & Media Information Gateway - one of the eLib-funded Internet information gateways.

Agora
An UK 'hybrid-library' project funded under Phase 3 of eLib to explore issues of distributed, mixed-media information management.

AHDS
Arts and Humanities Data Service - an UK service, funded by the JISC and the Arts and Humanities Research Board to collect, preserve and promote re-use of the electronic resources which result from research in the arts and humanities.

ANSI
American National Standards Institute.

Apache
An open-source HTTP server.

AQL
Aquarelle Query Language.

Aquarelle
An EU-funded project concerned with developing an information network for cultural heritage.

ARPA
Advanced Research Projects Agency.

ART
Proprietary format used by ARTISO - a gateway to the British Library Document Supply Centre's Automated Request Processing System (ARP) being developed by Fretwell-Downing Informatics. The gateway is compliant with the IPIG Profile for the ISO ILL Protocol.

ASF
Advanced Search Facility.

ASN.1 BER
Abstract Syntax Notation 1 Basic Encoding Rules.

ATHENS
An access management (authentication) service developed for and used by the UK higher education community that enables access to a variety of data-sets and information services.

BIOME
The RDN Hub for the health and life sciences.

Biz/ed
A Web-based service (including an Internet information gateway) for business and economics resources - one of the eLib-funded Internet information gateways.

Centroids
Index summaries. Used in the context of ROADS-based services to provide forward knowledge in a cross-searching environment.

CGI
Common Gateway Interface.

CHIC
See: TF-CHIC

CHIC-Pilot
A project developed by TF-CHIC that set up a pilot distributed indexing service based on WHOIS++, Harvest, ROADS and Z39.50.

CIMI Profile
A Z30.50 profile for cultural heritage information developed by the Consortium for the Computer Interchange of Museum Information (CIMI).

CIP
Common Indexing Protocol.

CNIDR
Center for Networked Information Discovery and Retrieval.

CNRI
Corporation for National Research Initiatives.

Combine
Software for harvesting and Internet resources - developed at NetLab as part of the DESIRE project.

COPAC
Bibliographic database - comprising a union catalogue of CURL member libraries' holdings.

CORBA
Common Object Request Broker Architecture.

CORC
Cooperative Online Resource Catalog. An OCLC initiative to build a union catalogue of Web-based electronic resource descriptions.

CURL
Consortium of University Research Libraries.

DanZig
Danish Z39.50 Implementers Group.

DAVIC
Design Audio Visual Council - an organisation responsible for creating specifications for end-to-end interoperability of broadcast and interactive digital audio-visual information, and of multimedia communication.

DC
Dublin Core.

DCOM
Distributed Component Object Model.

DCMI
Dublin Core Metadata Initiative.

DDC
Dewey Decimal Classification system.

DEF
Danmarks Elektroniske Forskningsbibliotek. Denmark's Electronic Research Library - a virtual library for researchers, students, lecturers and other users of Danish research institutions.

DESIRE
Development of a European Service for Information on Research and Education - a project funded by the European Union.

Dienst
A protocol and architecture for digital libraries that underlies NCSTRL.

DNER
Distributed National Electronic Resource - the JISC's concept of a managed environment for accessing heterogeneous 'quality assured information resources' on the Internet.

Dublin Core
An initiative - sometimes known as the Dublin Core Metadata Initiative (DCMI) - to develop a core metadata element set to facilitate the discovery of digital (networked) resources. Developments in the element set are defined on the basis of international consensus.

EELS
Engineering Electronic Library Sweden.

eLib
The Electronic Libraries Programme - a series of UK higher education-based networking projects, funded by the JISC.

Elki
Internet information gateway edited by the Library of Finnish Parliament.

EULER
European Libraries and Electronic Resources in Mathematical Sciences - a project funded by the European Union.

EUROPAGATE
A European Union-funded (Telematics for Libraries) project that developed a pilot gateway service through which different clients (including Web browsers) are able to access Z39.50 servers.

ETB
European Schools Treasury Broker.

EEVL
Edinburgh Engineering Virtual Library - one of the eLib-funded Internet information gateways. Now part of the EMC RDN Hub.

EMC
The RDN Hub for Engineering, Maths and Computing.

FDI
Fretwell-Downing Informatics.

FU
Functional Unit - a concept defined by the GAIA architecture.

FUM
Functional Unit Manager - a concept defined by the GAIA architecture.

FVL
Finnish Virtual Library.

GAIA
Generic Architecture for Information Availability - an EU-funded project aiming to provide a framework for multilateral information trading.

GEDI
Group on Electronic Document Interchange.

GILS
Global Information Locator Service.

GRS-1
Generalized Record Syntax One - a complex, general-purpose record syntax defined by the Z30.50 protocol.

Harvest
An open source software initiative offering a distributed solution to the problems of indexing data made available on the Web.

HBZ
The Online Utility and Service Center for Academic Libraries in North-Rhine Westphalia.

HTTP
Hypertext Transfer Protocol.

ht://Dig
A Web based indexing and searching package being developed as open-source software by a group of volunteers as a community led project.

IAFA
Internet Anonymous FTP Archive.

IDL
Interface Definition Language.

IETF
Internet Engineering Task Force.

IHR-Info
A gateway giving access to historical resources run by the Institute of Historical Research (IHR) of the University of London (since re-launched as HISTORY) - one of the eLib-funded Internet information gateways.

ILL
The ISO Interlibrary Loan protocols. There are two parts, a service definition (ISO 10160:1997), which defines the ILL services made available to applications using the protocol, and a protocol specification (ISO 10161-1:1997 and ISO 10161-2:1997), which specifies the content of protocol messages and the procedural rules for exchanging them.

IMesh
International Collaboration on Internet Subject Gateways - an international initiative with the aim of supporting communication and collaboration amongst subject gateway providers and related parties.

IMesh Toolkit
A project funded under the NSF/JISC International Digital Libraries Initiative to develop a configurable, reusable and extensible toolkit for subject gateway providers and to consider issues of relevance in the distributed, international subject gateway environment.

InterCat
Internet Cataloging project - OCLC project to test the use of the USMARC format (including the 856 field) and AACR2 cataloguing rules for describing Internet resources.

Internet Scout Project
Project located in the Computer Sciences Department at the University of Wisconsin-Madison providing summaries of selected high-quality Internet resources.

IPIG
ILL Protocol Implementors Group.

IRTF-RD
Internet Research Task Force Research Group on Resource Discovery.

Isaac Network
An initiative of the Internet Scout Project - linking selective collections of high-quality metadata-based Internet resources.

ISAD(G)
General International Standard Archival Description.

Isearch
Software for text indexing and searching - developed by CNIDR.

Isite
An integrated Internet publishing software package (including Isearch and Z39.50 communication tools) to access databases - developed by CNIDR.

ISO
International Organisation for Standardization.

ITC
International Electrotechnical Commission.

jake
Jointly Administered Knowledge Environment.

JISC
Joint Information Systems Committee - a committee funded by the Scottish Higher Education Funding Council, the Higher Education Funding Council for England, the Higher Education Funding Council for Wales and the Department of Education Northern Ireland. Its mission is 'to stimulate and enable the cost effective exploitation of information systems and to provide a high quality national network infrastructure for the UK higher education and research councils communities.'

Kilroy
An OCLC research project building an Internet harvester, full text databases, and metadata databases of Internet resources.

LCSH
Library of Congress Subject Headings.

LDAP
Lightweight Directory Access Protocol.

LDIF
LDAP Data Interchange Format.

MALVINE
Manuscripts And Letters Via Integrated Networks in Europe - a project funded by the European Union.

Mantis
A research toolkit developed at OCLC for building Web-based cataloguing systems.

MARC
MAchine Readable Cataloguing. A family of formats based on ISO 2709 for the representation and communication of bibliographic and related information in machine-readable form - e.g. MARC 21 or UKMARC.

MIA
MODELS Information Architecture.

MLO
Music Libraries Online - an UK 'clumps' project funded under Phase 3 of eLib creating a virtual union catalogue for music materials in British libraries, through a Z39.50 gateway.

MODELS
Moving to Distributed Environments for Library Services - an UKOLN initiative supported by JISC (through eLib) and the British Library.

MSC
Mathematics Subject Classification.

MySQL
A SQL database server produced by TCX DataKonsult.

NCSTRL
National Computer Science Technical Research Library.

NetFirst
OCLC service giving access to a database of Internet resource descriptions.

NISO
National Information Standards Organization.

NNDP
National Networking Demonstrator Project for Archives - a UK project commissioned by the JISC Non-Formula Funding (NFF) Archives Sub-committee to demonstrate and report on how Z39.50 and ISAD(G) might be successfully employed to provide multi-level cross searching of a range of nominated archival catalogues.

NNTP
Network News Transport Protocol.

NOVAGate
The Nordic Gateway to Information in Forestry, Veterinary and Agricultural Sciences.

NSF
National Science Foundation.

OCLC
Online Computer Library Center.

OMNI
Organising Medical Networked Information - one of the eLib-funded Internet information gateways. Now part of the BIOME RDN Hub.

Pavuk
A Unix-based program used to mirror contents of WWW documents or files.

PHP
An open-source, cross-platform, HTML-embedded scripting language used to create dynamic Web pages.

RDF
Resource Description Framework - a framework for metadata being developed by the World Wide Web Consortium (W3C) for a variety of different application areas, e.g. resource discovery, content ratings and intellectual property rights management. The *RDF Model and Syntax Specification* (Lassila and Swick, 1999) presents a syntax based on XML - sometimes known as RDF/XML.

RDN
Resource Discovery Network.

RDNC
Resource Discovery Network Centre - organisation responsible for co-ordinating the UK Resource Discovery Network, based jointly at UKOLN and King's College, London.

RIDING
An UK 'clumps' project funded under Phase 3 of eLib that aims to support large-scale resource discovery across the Yorkshire and Humberside region by using the Z39.50 protocol to create a distributed union catalogue.

ROADS
Resource Organisation and Discovery in Subject-oriented services - originally an UK project funded by JISC under eLib, ROADS is an open-source software toolkit for Internet subject gateways.

RTP
Real-Time Transport Protocol - designed within the IETF.

Scorpion
OCLC project building tools for automatic subject recognition based on well-known subject classification schemes.

SET
Secure Electronic Transaction - a protocol that facilitates secure payment card transactions over the Internet, promoted by the major credit card companies Visa and MasterCard.

SGML
Standard Generalised Markup Language - an international standard (ISO 8879) for the description of marked-up electronic text.

SOIF
Summary Object Interchange Format - a metadata format developed for use with the Harvest indexer.

SOSIG
Social Science Information Gateway - one of the eLib-funded Internet information gateways, now a RDN Hub.

SQL
Structured Query Language - a standard language for database applications.

TERENA
Trans-European Research and Education Networking Association.

TF-CHIC
Task Force-Cooperative Hierarchical Indexing Coordination - a TERENA-funded task force concerned with the co-ordination of harvesting and indexing networked resources.

UKMARC
The MARC standard developed and maintained by the British Library National Bibliographic Service (NBS).

UNIverse
EU-funded project - led by Fretwell-Downing Informatics -concerned with developing services for a distributed virtual union library service.

URN
Uniform Resource Name.

USMARC
The MARC standards maintained by the Library of Congress (Network Development and MARC Standards Office) - now harmonised with CAN/MARC (Canadian MARC) as the MARC 21 format.

WHOIS++
A search and retrieval protocol used, for example, by the ROADS software toolkit to ensure cross-searching.

WordSmith
OCLC project intended to improve user access to collections of electronic text by developing ways of identifying and organising clues about their content.

XML
Extensible Markup Language- a lightweight version of SGML developed for use on the Internet.

X.500
A set of International Telecommunications Union (ITU-T) standards covering electronic directory services (e.g. white page services).

YAZ
Yet Another Z39.50 Toolkit - a toolkit for implementing Z39.50 developed by Index Data.

Z39.50
An ANSI/NISO protocol for search and retrieval. Version 3 of the protocol has also been accepted as an ISO standard - ISO 23950.

Z39.50 EXPLAIN
A service added in version 3 of the Z39.50 protocol that allows a client to discover information about a server, such as available databases, supported attribute sets and record syntaxes.

ZAP
A search module for the Apache WWW server that utilises the Z39.50 protocol - developed in collaboration between Index Data and the US Geological Survey (USGS).

Zebra
A fielded free-text indexing and retrieval engine with a Z39.50 frontend developed by Index Data.

1 INTRODUCTION

Michael Day, UKOLN

The object of the Renardus project is to establish an academic subject gateway service in Europe. The pilot system will be based on a generic broker-architecture and data-model that will allow the integrated searching and browsing of distributed resource collections.

For reasons of easy extensibility, it is perceived that the development of a generic broker-architecture for Renardus will need to be based on a review of a variety of currently developed broker models. It is important to ensure that any chosen solution is based on emerging developments rather than being constrained by decisions made by the subset of gateways that are participating in the initial stages of the project.

Most existing broker models have been developed to solve particular solutions or to help provide certain services. For example, the ROADS software used by a number of Internet subject gateways has a model based on the use of the WHOIS++ protocol and the generation of index summaries (centroids) to enable cross-searching between multiple gateways. Other broker models have been developed to handle more complex requirements, including systems that broker access to a variety of different types of service types like Agora.

The broker models considered in this report represent a number of different architecture types:

- Generic architectures - e.g. the MODELS Information Architecture. MIA is used as a means of comparing the other architectural models reviewed in this report.
- Broker-type architectures developed for specific initiatives and projects - broadly speaking, these architectures broker access to a variety of different resource types, e.g. library catalogues, authentication servers, etc. These include projects like Agora, Aquarelle, EULER, GAIA and UNIverse.
- Architectures developed to enable the cross-searching of distributed Internet information gateways - usually based on the same search and retrieval protocol, e.g. WHOIS++ or ANSI/NISO Z39.50 (ISO 23950). Examples include, e.g. the architectures that underlie ROADS cross-searching, the Resource Discovery Network ResourceFinder and the Finnish Virtual Library.

This report attempts to review and evaluate a number of these existing broker architectures to ensure that the Renardus architecture is an example of best practice and that work is not unnecessarily duplicated.

1.1 Broker services

One of the biggest challenges facing those who are attempting to develop digital libraries at the present time is attempting to integrate access to the wide range of distributed and heterogeneous information resources and services that are available. The successful integration of these resources and services is perceived as of being of great benefit to libraries and their end users. Dempsey, Russell and Murray (1999, p. 35) point out that resources are typically differently presented, accessed and structured, and that users, for example, may have to interact with a number of quite different information systems in order to carry out a full search. They suggest the development of an additional service layer - here described as 'middleware' - that would shield the user from any underlying complexity and heterogeneity. This middleware - a broker service - would need to provide "a higher level interface, creating a federated resource from underlying heterogeneity and mediating access to it" (Dempsey, Russell and Murray 1999, p. 38).

Most broker development relates to particular projects or services, e.g. the development of a distributed and heterogeneous mathematics information service in project EULER, or the RDN ResourceFinder. Despite this, several projects and initiatives have tried to address more generic issues. For example, the development of generic broker infrastructure was one of the objectives of the Stanford Digital Library project - funded as part of the original US Digital Libraries Initiative. This project, based at Stanford University, developed a modular testbed infrastructure known as an information bus (or Infobus) based on CORBA (the Common Object Request Broker Architecture) that enabled the integration of a variety of different digital library functions (Baldonado, et al., 1997; Paepke, et al., 1996, Paepke, et al., 1999).

In the UK, much work has been mediated through the MODELS initiative and the development of a generic MODELS Information Architecture.

1.2 The MODELS Information Architecture (MIA)

The MODELS (MOving to Distributed Environments for Library Services) project is an UKOLN initiative that has gained additional support from JISC (through the Electronic Libraries Programme) and the British Library, with Fretwell-Downing Informatics (FDI) as technical consultants. MODELS provides a forum - primarily directed through a series of workshops - that allows relevant stakeholders to explore shared concerns about distributed and heterogeneous resources and services. The initiative has attempted to address design and implementation issues, initiate concerted actions, and work towards a shared view of preferred systems and architectural solutions. It has also played a major role in the development of policy and emerging services in the UK.

MODELS-facilitated deliberations have led to the development of a logical framework for information management in a distributed environment known as the MODELS Information Architecture (MIA). This is not a specification for any particular implementation but a generic model intended to support the discussion and comparison of alternative solutions. Dempsey, Russell and Murray (1999, pp. 38-39) describe the function of MIA in the following way:

> *The MIA is aligned with wider work that sees the development of 'middleware' or 'broker' services as a central part of how the information environment will develop. It is concerned with the types of function such 'broker' services need to provide as they help project a unified service over a distributed, heterogeneous set of network services. It has a dual focus: as a conceptual heuristic tool for the library community which helps clarify thinking and acts as a lever for development, and as a tool to assist developers as they think about future systems work. The main emphasis has been on the former aspect. The MIA investigates the functional components of viable digital information environments and arranges them in a logical architecture: it does not yet specify how components will be implemented, or concrete interfaces.*

The MIA has been described in more detail in a couple of documents produced as part of an MIA Requirements Analysis Study carried out by UKOLN. The first of these describes the MIA's logical architecture, the second its functional model.

The MIA Functional Model (Gardner, Miller and Russell, 1999b) defines the user functionality of a hybrid information environment. This assumes that the basic behaviour of a hybrid information system is associated with the four MODELS functions:

- Discover
- Locate
- Request
- Deliver

The MIA logical architecture is a layered architecture with five layers. The following descriptions of these layers are adapted from the draft paper by Gardner, Miller and Russell (1999a).

- **Presenter** - This layer is responsible for interacting with users (both human and software), i.e. presenting information to, and accepting input from, the user. The presentation layer may need, for example, to generate HTML, a non-Web GUI interface, a spoken interface, a command-line interface, an email-interface or whatever is required in a particular application. Software clients may also need to be catered for - providing, for example, Z39.50, WHOIS++ and LDAP interfaces to the system. Information entered by the user must be passed on to the Coordinator; this may involve creating 'objects' to pass through agreed APIs or it may involve encoding the data in a particular transport format, e.g. XML.
- **Coordinator** - The Coordinator provides an application layer on top of the Mediator. The Coordinator may provide high-level services built on lower-level services provided by the Mediator appropriate to its user community (e.g. to search for local resources) and may add offer value-added services to its user community, such as maintaining bookmarks and providing services tailored according to user profiles. The Coordinator is responsible for application logic including user profiles (where we include the possibility that the user may be a human or a software agent) and session maintenance (which is concerned with taking

into account previous actions within the current session). The Coordinator contextualises requests to the current situation.

- **Mediator** - The Mediator is responsible for understanding the meaning of services (such as search, locate, request and deliver) that may be offered by providers and requested by the Coordinator. The Mediator receives requests from the Coordinator and must determine which service providers (in parallel or in combination) can satisfy the request. Requests may be complex - e.g. a search for a particular book and then to locate libraries that hold the book (and possibly online bookstores that sell it).
- **Communicator** - The Communicator is responsible for communicating with external services, it shields the Mediator from details such as communication protocols and service locations and may also provide basic mapping between metadata vocabularies to achieve a vocabulary understood by the Mediator. In some cases services may directly support communication with the Mediator (they may have been developed to be compatible with it), in such cases the action carried out by the Mediator will be trivial. The Communicator provides a gateway between the Mediator and providers based on a Network Service Profile associated with each service associated with a provider. The Network Service Profile provides details of the location, protocol, query and response formats and metadata vocabularies that required in order to meaningfully access a service.
- **Provider** - The Provider layer contains the external services accessed by the system. The layer most obviously includes the 'primary' services for which the system exists to provide access; for example library catalogues, abstracting services and subject gateways. The provider layer also includes 'secondary' services that the system must access in order to provide primary services, for example, schema registries, authentication services and user profile directories. It is recommended that all services that could be shared with other systems be externalised in this way.

As stated earlier, MIA is not a specification for any particular implementation but a generic model intended to support the discussion and comparison of alternative solutions. The model has, however, been used as the basis for the development of a number of implementations. For example, the UK Arts and Humanities Data Service (AHDS) 'resource discovery' system (Beagrie, 1999) is based on the layered approach to resource discovery developed as part of the MODELS initiative (Russell, 1997), and is broadly based on MIA concepts (Greenstein and Murray, 1997). MIA has also influenced the development of UK hybrid library projects like Agora, services like the Resource Discovery Network. It has also underpinned the JISC's idea of a Distributed National Electronic Resource (DNER). So, for example, a simplified MIA structure has been developed by Powell (1999) to describe DNER services known as DNER Portals (Figure 1.1). It is this structure that has been used as a basis for comparing the different broker models considered in this review.

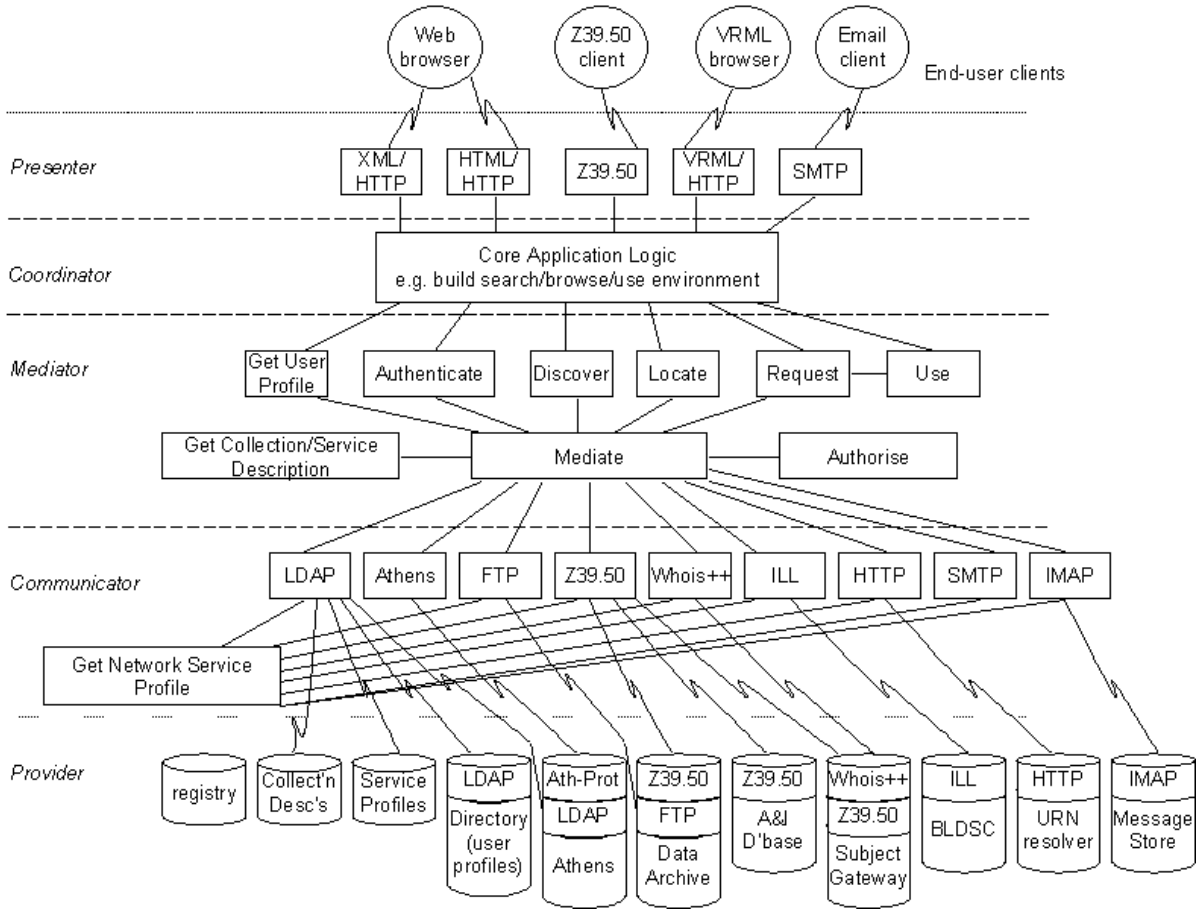


Figure 1.1: MIA view of a DNER Portal (Powell, 1999)

2 THE BROKER REVIEW

The 18 broker models described here are reviewed with reference to the MIA. Each broker architecture is briefly introduced and then the Presenter, Coordinator, Mediator, Communicator and Provider levels are described in more detail, sometimes with the help of a diagram based on the MIA view of a DNER Portal (Fig 1.1).

Martin Hamilton of the Department of Computer Science at Loughborough University of Technology (LUT) is carrying out a similar review of broker models for the IMesh Toolkit project - as part of the IMesh Toolkit Architectural Overview deliverable. Some of these reviews (ASF Freeware, CHIC-Pilot, Harvest, ht://Dig, Isaac Network, jake, ROADS) have - with permission - been adapted for inclusion in this report. Other reviews have been provided by Renardus project partners and by Matthew J. Dovey of the University of Oxford Libraries Automation Service (LAS).

Note that broker architectures (including GAIA) have also been evaluated in a report undertaken as part of the EU-funded MALVINE project (Langer, Adametz and Fellien, 1999).

2.1 Agora

Andy Powell, UKOLN

2.1.1 Introduction

2.1.1.1 Responsible agency

Agora is a project funded by JISC under phase III of the Electronic Libraries Programme (eLib). The University of East Anglia leads the project, with UKOLN, Fretwell-Downing Informatics (FDI) and the Centre for Research in Library in Information Management (CERLIM) at Manchester Metropolitan University as the other partners.

2.1.1.2 Description/Scope

Agora is a 'hybrid-library' project in that it attempts to integrate the technologies developed for new digital services with those used to give access to traditional library collections. The project builds upon work carried out within MODELS - especially the MIA - and is developing a Hybrid Library Management System (HLMS) that will be an MIA-type broker - a hybrid library demonstrator that provides discover, locate and request functionality across a range of resources. Through this broker, the project is experimenting with providing integrated access to a variety of services that use different protocols and have different interfaces, including library catalogues, Web index services, information gateways and document supply services.

2.1.1.3 Architectural diagram

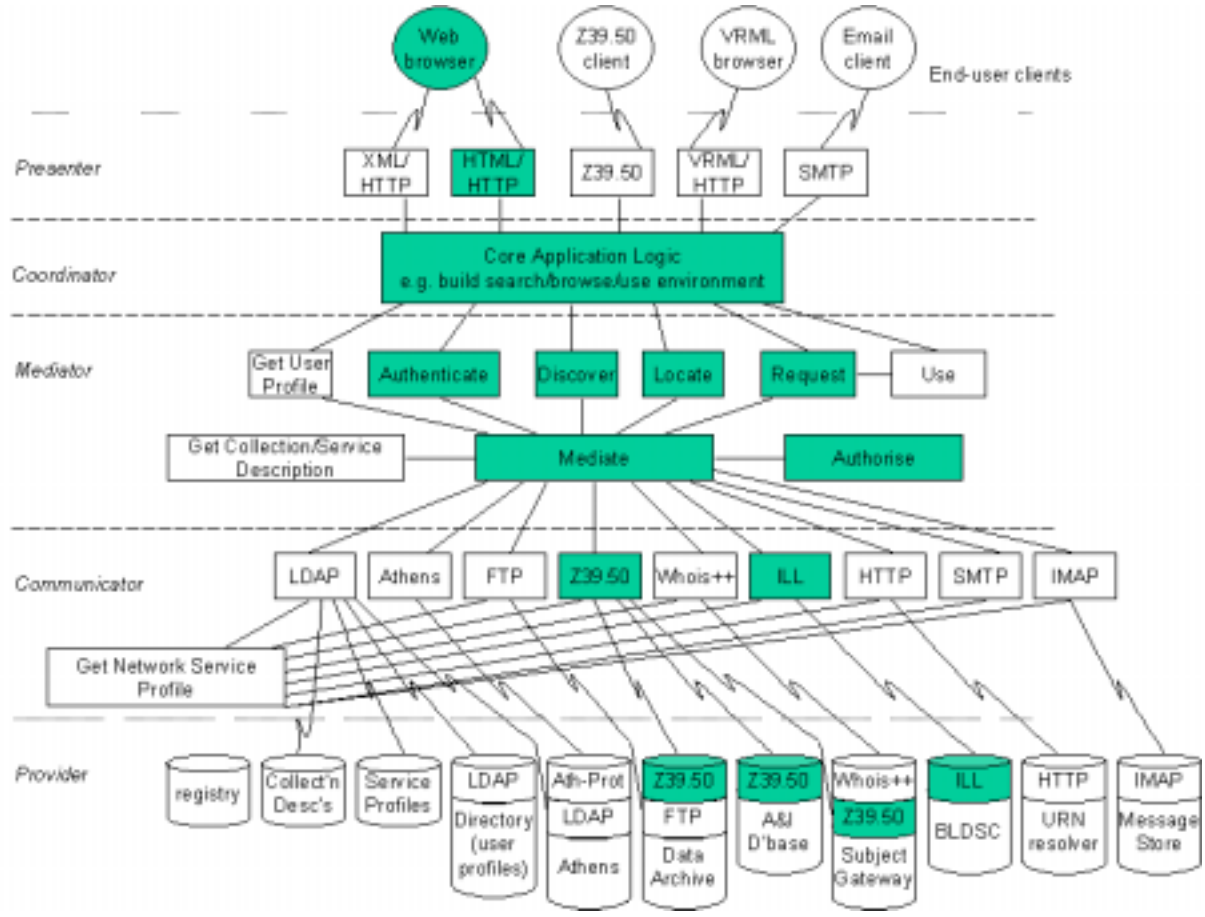


Figure 2.1: Agora related to MIA

2.1.2 Logical architecture (MIA framework)

2.1.2.1 Presenter

The Agora presenter layer provides an HTML/HTTP interface to the Agora 'broker'. A Z39.50 target is expected to be supported in the next version of the Agora presenter layer.

2.1.2.2 Coordinator - Mediator - Communicator

The Agora coordinator, mediator and communicator layers support discover, locate and request functions through underlying use of the Z39.50 and ILL protocols. Authentication, collection description and service profiles are currently stored internally within the Agora system.

Agora expects to provide support for ATHENS-based authentication in the next version of the software.

2.1.2.3 Provider

The Agora provider layer implements a Z39.50 client, through which the mediator communicates with target Z39.50 servers, and an ILL client. It supports several protocols, including ILL and ART, to enable end-users to request delivery of resources (though this is partly not working currently because of the lack of an ILL gateway to the British Library Document Supply Centre).

2.1.3 Technical architecture

2.1.3.1 Standards

Collection description is based on the schema developed jointly by several eLib phase 3 projects (including RIDING, Agora and MLO). The schema incorporates and extends the Dublin Core.

Agora supports the Generic Document Interchange (GEDI) standard for electronic delivery of resources.

2.1.3.2 Protocols

Z39.50 and the Bib-1 attribute set. The following record types are handled by Agora:

USMARC, UKMARC, Dublin Core, Digital Collections - NNDP Profile, Universe Clusters, GRS-1 generic.

Agora is currently considering implementing support for the California Digital Library protocol.

2.1.3.3 Software

Fretwell-Downing Informatics (FDI) developed the Agora Hybrid Library Management Systems based on their OLIB VDX system.

2.1.4 References

Agora project: http://hosted.ukoln.ac.uk/agora/

There is also a short description of Agora in Dempsey, Russell and Murray (1999, pp. 58-59).

2.2 Aquarelle

Matthew J. Dovey, LAS

2.2.1 Introduction

Aquarelle was a project funded by the European Commission under its Telematics Applications Programme from 1996-1998 (Michard, 1998). The project was concerned with developing an Information Network on Cultural Heritage. This is a distributed information system that attempts to provide uniform access to the varied collections of data held by museums, art galleries and other cultural heritage-based organisations within Europe.

2.2.1.1 Responsible agency

The Aquarelle project, a European consortium of cultural heritage organisations, IT companies, publishers and research organisations, managed by ERCIM, the European Research Consortium for Informatics and Mathematics. More information is available on the INRIA Web site:

http://aqua.inria.fr/Aquarelle/

2.2.1.2 Description/Scope

Aquarelle is a project designed to provide access to museum collections. As such it is designed to provide a resource discovery system for cultural heritage information. It recognises that archives typically store information in both:

- Existing primary material called *archive data* - e.g.: records, drawings or maps. Typically, these are available in a variety of forms and structures, e.g.: text, images, databases, etc.
- Specially created secondary material that consists of structured, hierarchical (e.g. SGML-based) documents that describe, comment on and refer to archive data, etc. Aquarelle developed a mechanism (called *folders*) for searching and navigating the latter.

The project also provides image-watermarking facilities.

2.2.1.3 Architectural diagram

The architecture of Aquarelle is given below. Access to the underlying databases (both archival and folder databases) is provided via a Z39.50 gateway to the underlying database.

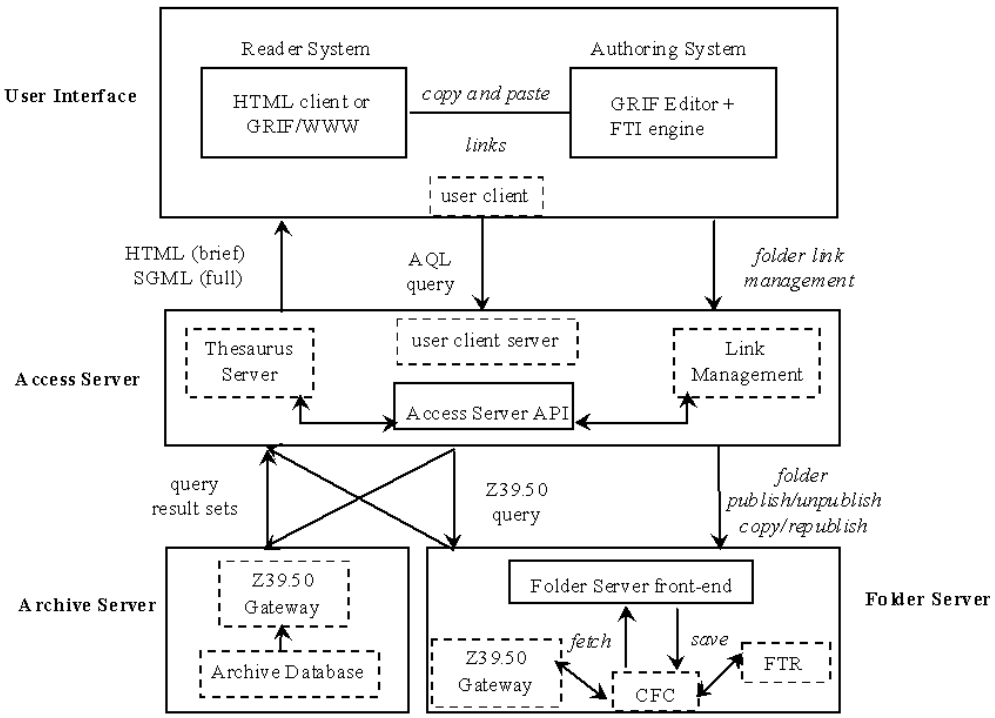


Fig. 2.2: The Aquarelle Architecture (Michard, 1998)

2.2.2 Logical architecture (MIA framework)

Presenter Dempsey, Russell and Murray (1999, p. 64) point out that the Aquarelle system can be readily mapped to the MIA model:

> *The Aquarelle Access Server is a broker supporting most of the MIA functions. It maintains a database of registered users for authentication and user profiles. The Aquarelle Directory services maintain a database of collection and service descriptions that support the MIA discover and locate functions. The user interface components present the information landscape in terms of subject domains as well as specific databases; they also provide multilingual thesauri to assist in query formulation. Aquarelle supports the MIA request and delivery functions for Aquarelle folders. In addition Aquarelle offers facilities which are not explicit MIA components: folder publishing, persistent link management and multilingual thesauri for query formulation.*

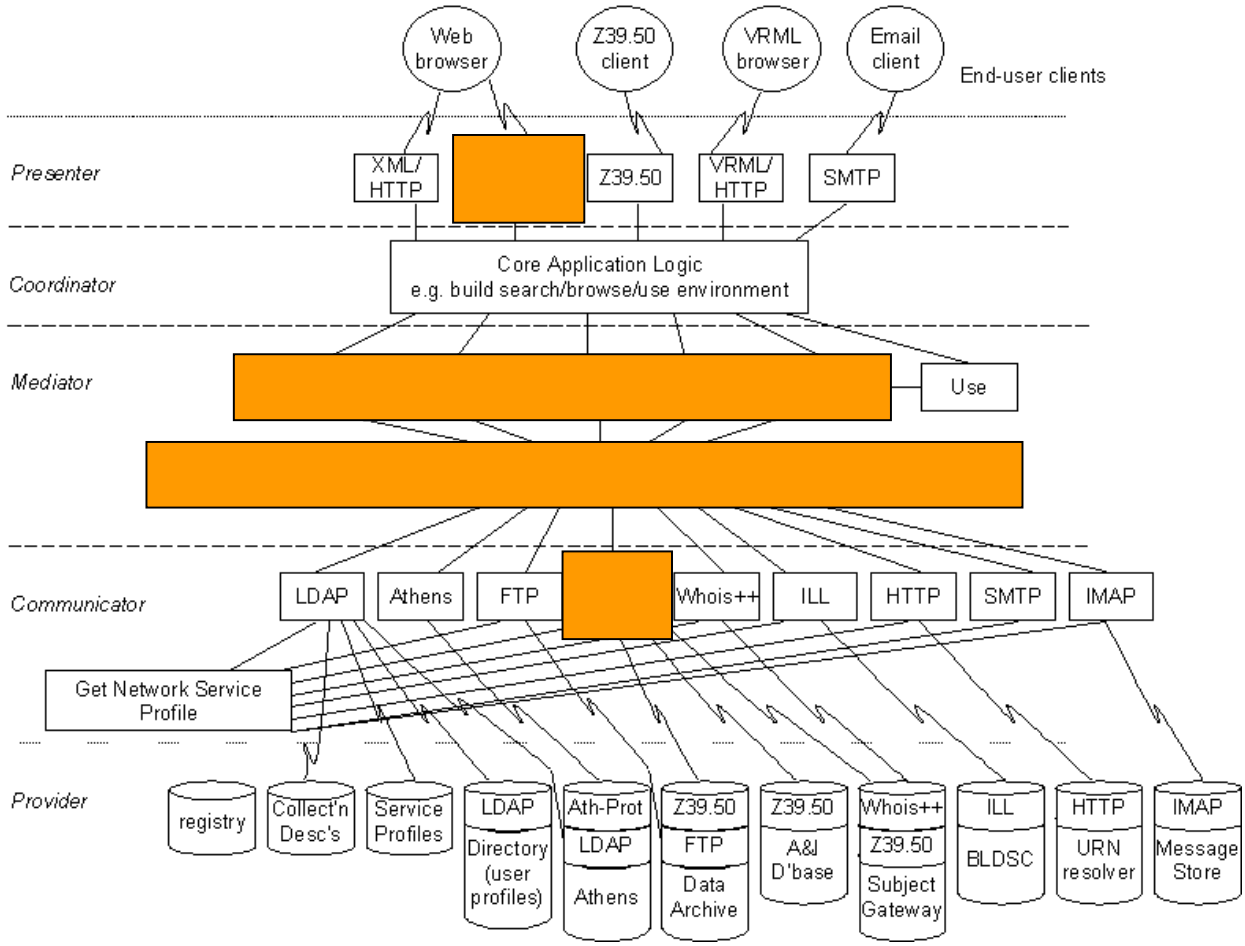


Figure 2.3: Aquarelle related to MIA

2.2.2.1 Presenter

The presenter in the Aquarelle model is the user interface layer. This presents information to the end user via HTML over HTTP. It communicates to the user client server in the Access Server layer of the Aquarelle model using an internal query language AQL (Aquarelle Query Language)

2.2.2.2 Coordinator

The co-ordinator in the Aquarelle model is provided by the user client server in the Access Server layer. The user client server also provides some mediator facilities such as authentication and user profiling. There is also an addition function provided at this layer in the form of a thesaurus management system and a link management system for navigating structured data (in the form of SGML). More information on the link management system can be found in: E. Fras, *Folder Publisher, Link Server and Directory Services*, Aquarelle Deliverable No D4.3, 15 Sept. 1997.

2.2.2.3 Mediator

The mediator in the Aquarelle model is provided partly by the Access Server API in the Access Server layer, but some facilities as mentioned above are provided by the user client server.

2.2.2.4 Communicator

Communications from the mediator to the underlying providers is via the Z39.50 protocol using a profile incorporated into the CIMI profile. Records are returned via GRS-1 syntax encoded SGML records.

2.2.2.5 Provider

The provider needs to be a Z39.50 server. Either this is a dedicated Aquarelle *folder* server providing access to SGML hypertext documents typically providing information relating to groups of objects and including links to other folders, and to object data known to the Aquarelle system, or to an existing record database. In the case of the latter this may be provided via a Z39.50 gateway for that particular database.

2.2.3 Technical architecture

2.2.3.1 Standards

Aquarelle developed a Z39.50 Application Profile based on Draft version 3 of the CIMI profile. This is now merged into the CIMI standard. Records are transmitted as GRS-1 encoded SGML records. The thesaurus management system is based on the standards ISO 2788 and ISO 5964.

2.2.3.2 Protocols

The communication protocols are Z39.50 and HTTP. There is also an internal query language AQL (Aquarelle Query Language).

2.2.4 References

Aquarelle: http://aqua.inria.fr/aquarelle/public/EN/home-eng.html

Michard, A., ed., 1998, *Final report: IE-2005 Aquarelle: sharing cultural heritage through multimedia telematics*. Le Chesnay: INRIA. http://aqua.inria.fr/Aquarelle/Public/EN/final-report.html

Michard, A., Christophides, V., Scholl, M., Stapleton, M., Sutcliffe, D., Vercoustre, A-M., 1998, The Aquarelle resource discovery system. *Computer Networks and ISDN Systems*, 30(13), 1185-1200.

Z39.50 for Access to Cultural Heritage Information - Aquarelle Profile, Version 0.2, 1997-07: ftp://lcweb.loc.gov/pub/z3950/profiles/aqua.txt

2.3 ASF Freeware

Martin Hamilton, LUT

2.3.1 Introduction

2.3.1.1 Responsible agency

ASF (Advanced Search Facility) is an interoperability framework for (predominantly) government information. The 'freeware' ASF implementation reviewed here is being developed by a diverse group that was initially funded under the U.S. Information Technology Innovation Program.

2.3.1.2 Description/Scope

The ASF freeware distribution consists of:

- ASFcrawl - a robot based indexer (the Pavuk package)
- ASFserv - a Z39.50 server (built with the Index Data YAZ library)
- ASFhttpd - a Z39.50 enabled HTTP server (a custom Apache server with the Index Data 'ZAP' Z39.50 client module, which also uses YAZ)

There are also a number of CGI based and X Window admin programs for configuring and controlling the ASF node, and an experimental WHOIS++ server.

The underlying search engine for ASFserv and ASFwhois is the CNIDR Isearch library.

2.3.2 Logical architecture (MIA framework)

2.3.2.1 Presenter

The ZAP Apache module

2.3.2.2 Coordinator - Mediator - Communicator

N/A

2.3.2.3 Provider

The ASFserv Z39.50 server

The ASFcrawl component of the ASF Freeware package effectively constitutes an additional layer that MIA does not directly address.

2.3.3 Technical architecture

2.3.3.1 Standards

The end user interacts via HTTP with an Apache module mod_zap that provides the actual search capabilities. The database being searched is constructed using ASFcrawl, which performs HTTP based traversal and indexing starting at a nominated URL. The ASF database is normally accessed via the Z39.50 search and retrieval protocol, but experimental WHOIS++ access is also available.

2.3.3.2 Protocols

The core protocols used by the ASF Freeware distribution are HTTP and Z39.50, with GILS records encoded in XML as the standard metadata format. Support for WHOIS++ and RFC 1913 centroids - but not the Common Indexing Protocol (RFC 2651, RFC 2652).

2.3.3.3 Software

No additional software is required to get a server up and running with the ASF Freeware package.

Version Reviewed: 1.3.2 (17th March 2000)

Download: http://asf.gils.net/

Status: A number of packages each with their own conflicting copyrights, plus additional code written by the ASF group which has no copyright assignment.

Support: Community support is available via the asf@cni.org mailing list.

Platforms: Development and testing are done using Linux. Some support for other Unix variants (e.g. OpenBSD and NetBSD) are also available.

Prerequisites: Linux, Perl, C/C++ compiler

2.3.4 Conclusions

The ASF Freeware package is essentially a Web crawler, like ht://Dig, and so not immediately suitable for use as in the subject gateway context. Unlike many of the other robot based indexing packages, it features a standard metadata format, which would make it possible to introduce records produced by human cataloguers in addition to those generated by the robot based indexing process.

Similarly, use of Z39.50 and WHOIS++ servers means that the ASF Freeware package can be integrated into existing Z39.50 and WHOIS++ based systems.

Since the Z39.50 server sits in between the end user and the system (in normal operation), there may be problems with high volume usage. There is no indication that this has been tested, and no high volume sites are known to be using the ASF Freeware package.

As with early versions of Harvest, complete external packages (Apache, Pavuk, Isearch and ZAP) are included - rather than having the installer fetch them separately.

This is unfortunate, since it creates all manner of problems for the ASF software maintainers, and for anyone wishing to extend the software. It is also misleading, since the ASF Freeware developers (in their announcements and release notes) effectively claim credit for this work, which has been done by other people. Only a tiny fraction of the ASF Freeware distribution is actually new code. To their credit, the ASF developers have taken pains to keep their versions of Apache et al. separate and distinct by keeping them within the ASF directory structure.

The use of shell scripts in the ASF CGI based admin system is a major cause for concern on security grounds - but in normal operation this area is only accessible to people with admin privileges on the server. Anyone operating an ASF server should take care who they grant admin privileges to, and the scripts themselves should be rewritten - e.g. in Perl with tainted variable checking.

The ASF Freeware distribution shows us how existing packages may readily be combined to form a complete resource discovery system. However, the major areas of functionality in (for example) ROADS are not represented - e.g. Web based editing, link checking, 'what's new' and subject listing breakdowns.

2.4 CHIC-Pilot

Martin Hamilton, LUT

2.4.1 Introduction

TF-CHIC (Task Force- Cooperative Hierarchical Indexing Coordination) was a TERENA-funded task force concerned with the co-ordination of harvesting and indexing network resources. Work in the task force built upon existing standards and technologies, such as those employed in Harvest and the DESIRE and ROADS projects.

The task force spawned the CHIC-Pilot project that set up a pilot distributed indexing service based on WHOIS++, Harvest, ROADS and Z39.50 technology. The project ran from the end of 1997 to the Summer of 1998, and its results were later fed back into the ROADS software development.

2.4.1.1 Responsible agency

TF-CHIC, a task force co-ordinated and funded by the Trans-European Research and Education Networking Association (TERENA).

2.4.1.2 Description/Scope

The CHIC-Pilot architecture is specifically geared up towards distributed indexing - to the extent that it incorporates a layer of functionality (the Gatherer) specifically for this. By contrast, the MIA model does not concern itself with the origin of the data offered by the Provider to the other layers.

CHIC-Pilot never managed to get to the point of using centroids from the various Indexers included - instead a simple hack was used whereby the chic-search.pl script connected to a WHOIS++ server which returned referrals to each of the WHOIS++ servers (including application level gateways) which it was aware of.

2.4.1.3 Architectural diagram

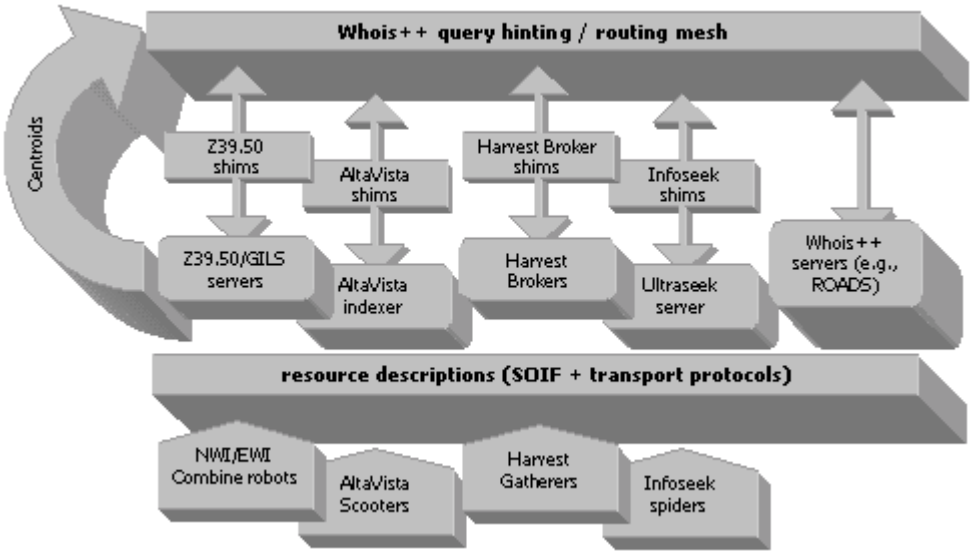


Figure 2.4: CHIC-Pilot overview (Valkenburg, et al. 1998)

2.4.2 Logical architecture (MIA framework)

The CHIC-Pilot architecture is based on three layers:

- Broker (CGI script chic-search.pl) - is responsible for interacting with the end user, sending their queries to the Indexer using WHOIS++ (RFC 1835), and collating the results.

- Indexer (various) - collects resource descriptions from the Gatherer, indexes them, and makes the indexes available for searching by the Broker. The Indexer is required to implement the WHOIS++ protocol, and may also support WHOIS++ centroids (RFC 1913) or Common Indexing Protocol (RFC 2651) Tagged Index Objects (RFC 2653).

- Gatherer (various) - creates resource descriptions by visiting URLs and summarising their contents, and makes these available for fetching by the Indexer. The Gatherer is required to support the SOIF [harvest] metadata format for information exchange with the Indexer.

A fuller description of the CHIC-Pilot architecture is available in a paper which was presented at the 1998 TERENA Networking Conference (Valkenburg, et.al., 1998).

Note that although CHIC-Pilot was conceived for distributed indexing and searching based on `robot' generated metadata, this is not actually a requirement. The metadata that is manipulated could have been created by a human cataloguer rather than by machine parsing.

2.4.2.1 Presenter - Coordinator - Mediator - Communicator

The top four layers of MIA are essentially compressed into the CHIC-Pilot Broker. This is feasible because of the use of lowest-common-denominator 'standards' in the form of WHOIS++ and SOIF, and means that elaborate mechanisms to incorporate support for multiple standards in these areas are not necessary in the architecture itself. However, as a result it would be non-trivial to replace either of these standards.

2.4.2.2 Provider

Native WHOIS++ servers or gateways to other systems (e.g. UltraSeek)

2.4.3 Technical architecture

2.4.3.1 Standards

The CHIC-Pilot architecture is built around HTTP and HTML for interaction with the end user.

2.4.3.2 Protocols

WHOIS++ is used for interaction between the Broker and the Indexers, and SOIF for exchanging index information between the Indexer and the Gatherer.

2.4.3.3 Software

Although no additional software beyond *chic-search.pl* is required in order to operate the CHIC-Pilot search interface, this pre-supposes that a network of WHOIS++ servers are available to query. To use the WHOIS++ gateways to Harvest, Z39.50 and UltraSeek, the operator must have these servers running behind the scenes.

The script *chic-search.pl* is open source software distributed under the terms of the GNU General Public License/Perl Artistic License (i.e. the standard Perl Terms and Conditions).

2.4.4 References

More information on the CHIC-Pilot (and a cross-search demonstration) can be found at: http://www.terena.nl/projects/chic-pilot/

Valkenburg , P., Beckett , D., Hamilton, M., Wilkinson , S., 1998, *Standards in the CHIC-Pilot Distributed Indexing Architecture*. TERENA Networking Conference '98, Dresden, 7 October. http://www.terena.nl/projects/chic-pilot/tnc/paper.html

2.5 Cooperative Online Resource Catalog (CORC/Mantis)

Arthur N. Olsen, NetLab

2.5.1 Introduction

The OCLC Cooperative Online Resource Catalog (CORC) builds on the experience of earlier OCLC initiatives like NetFirst and InterCat. CORC is based on the same philosophy as the main OCLC union catalog (WorldCat). Guest access to CORC is available at the following URL: http://www.oclc.org/oclc/corc/index.htm

2.5.1.1 Responsible agency

The Online Computer Library Center (OCLC)

2.5.1.2 Description/Scope

CORC is primarily a union catalogue of Web-based electronic resource descriptions that parallels WorldCat. The system includes a number of innovative features to assist in cataloguing new Web sites or pages. The system provides harvested documents with suggested subject keywords and classification numbers (DDC). Link checking is provided to ensure the currency of URLs and automated content checking assists in determining the stability of the resource. The participating libraries can use the system to construct and maintain lists of Web resources in specific areas - called *Pathfinders* in OCLC parlance.

Description in CORC can be done in both MARC21 and DC formats, data can be imported and exported in MARC or RDF/XML format. Internally the bibliographic data is held in ASN1.BER format.

The size of the CORC database is currently (March 2000) about 200,000 records. Background information about CORC has been mostly published in the *OCLC Newsletter* (OCLC, 1999)

2.5.1.3 Architectural diagram

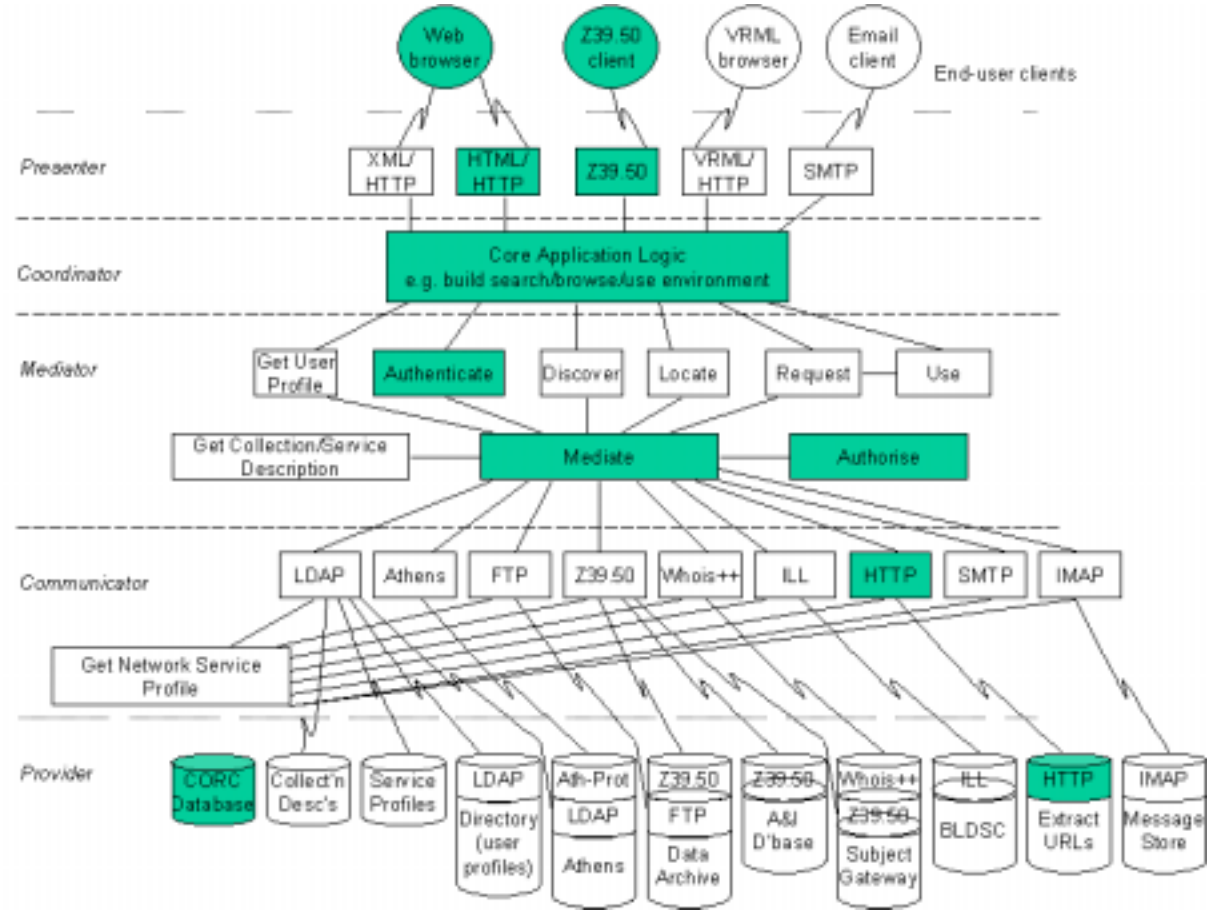


Figure 2.5: CORC related to MIA

2.5.2 Logical architecture (MIA framework)

2.5.2.1 Presenter

User access is with standard Web browsers. Z39.50 access is possible for searching

2.5.2.2 Coordinator

Core logic for searching, cataloguing and constructing pathfinders. Also functions for automated subject indexing and classification of electronic documents.

2.5.2.3 Mediator

Mediator functionality is rather limited. CORC is a union catalogue concept with a large degree of centralisation. It is reasonable to place authorisation in this layer.

2.5.2.4 Communicator

The amount of communication with other systems is limited, retrieving resources for description and harvesting is based on HTTP.

2.5.2.5 Provider

The information provided is located in the central CORC database

2.5.3 Technical architecture

2.5.3.1 Standards

Metadata in MARC 21 and DC format, export in RDF/XML format is also possible. Internal representations in ASN1.BER format.

Subject description standards based on Dewey Decimal Classification (DDC) and Library of Congress Subject Headings (LCSH).

2.5.3.2 Protocols

HTTP and Z39.50 protocols

2.5.3.3 Software

Most of the software used in the CORC system has been developed at the OCLC Office of Research. The MANTIS toolkit has been used as a basis for constructing CORC. The software tool Kilroy is used for harvesting Internet resources. Two software tools are used for assisted subject indexing of Web documents, Scorpion for Dewey (DDC) numbers and WordSmith for subject keywords. All the software mentioned is proprietary. *Licensing Scorpion with access to the Dewey database could be an option regarding provision of a common browsing framework for a future Renardus service*.

2.5.4 References

CORC: http://www.oclc.org/oclc/corc/index.htm

MANTIS: http://orc.rsch.oclc.org:6464/

Kilroy: http://orc.rsch.oclc.org:7080/

Scorpion: http://orc.rsch.oclc.org:6109/

Wordsmith: http://orc.rsch.oclc.org:5061/

Hickey, T.B., 2000, CORC: a system for gateway creation. *Online Information Review*, 24 (1), 49-53.

OCLC, 1999, CORC Project. *OCLC Newsletter*, 239, May/June 1999. http://www.oclc.org/oclc/new/n239/index.htm#feature

2.6 DEF - Denmark's Electronic Research Library

Anders Ardö, DTV

2.6.1 Introduction

2.6.1.1 Responsible agency

The Danish Ministry of Culture, the Danish Ministry of Research and the Danish Ministry of Education have decided together to carry out the project 'Denmark's Electronic Research Library' (DEF). The National Budget for 1998 provided a total amount of 200 mil. DKK for the project, distributed over the years 1998-2002.

DEF is established through a network of research libraries, information centres and public libraries with a view to creating Denmark's Electronic Library.

It is important to stress that DEF should emerge as one large, coherent, virtual information system as a result of the network's linking of the research libraries' and other information centres' services, including for example national licence agreements. The overall effect is gained by complying with and following standards: for communication, for search support, for terminology, for registration of document description and representation.

2.6.1.2 Organisation

The Liaison Committee works out the framework for DEF and consists of representatives for the three ministries and the Danish National Library Authority as well as the chairman of the Steering Committee. The Steering Committee consists of 10 representatives for relevant user groups. They have the overall responsibility for turning the project into reality. The Danish National Library Authority acts as secretariat to the Steering Committee and is responsible for the actual running of the project.

Denmark's 12 largest and 44 medium-sized research libraries and the Danish National Library Authority are the primary forces in the DEF project. But in due course more than 200 small research libraries and the country's other information suppliers an research institutes will become part of DEF.

2.6.1.3 Description/Scope

There are three main architectural components in DEF:

1 **DEFkey** - a global authorisation system that should ensure access to all relevant DEF services, irrespectively of location, through one login action (still under investigation/development).

2 **DEFcat** - a virtual union library catalogue of all participating research libraries individual catalogues. It is based searching each catalogue using Z39.50 and the common DanZig profile (in testing phase).

3 **DEFportal** - gateway to Internet resources. This component is the most developed and it is (presently) made up of 6 individual services, where all are using similar metadata profiles but covering different subject areas:

- Vejviser - Covers resources at Danish research libraries. Based on metadata in Web pages that are harvested to produce a central database once a week (in production since 1999-10-01).
- Industrial economics, descriptive statistics - a consortium with 4 members
- Virtual music library - a consortium with 5 members
- Medical clinical information - a consortium with 4 members
- Food technology - a consortium with 4 members
- Energy technology - a consortium with 3 members

Most consortia members are research libraries. The subject specific portals (2-6) have just started their work and are planned to go public by the end of 2000. They base their metadata definitions on the basic Vejviser profile.

All of the above (1-6) also provides, in addition to resource descriptions according to the metadata format, a full-text database of harvested Web pages relevant for each of the services.

2.6.2 Architectural diagram

The basic architecture relies on underlying structured data being provided through a standardized search and retrieval protocol (Z39.50) with a few basic profiles (DanZig, Portal). On top of that user interfaces are/can be built using either Web to Z39.50 gateways or native Z39.50 clients. Discussions and implementations about user interfaces include:

- virtual union catalogues
- vejviser
- each subject specific portal individually
- vejviser and all subject portal in a seamless union
- the above plus virtual union catalogue
- other more advanced integrating more sources like article databases (DADS), Web indexes, etc.

2.6.3 Logical architecture (MIA framework)

Here is the planned seamless service brokering the 6 portal services is described.

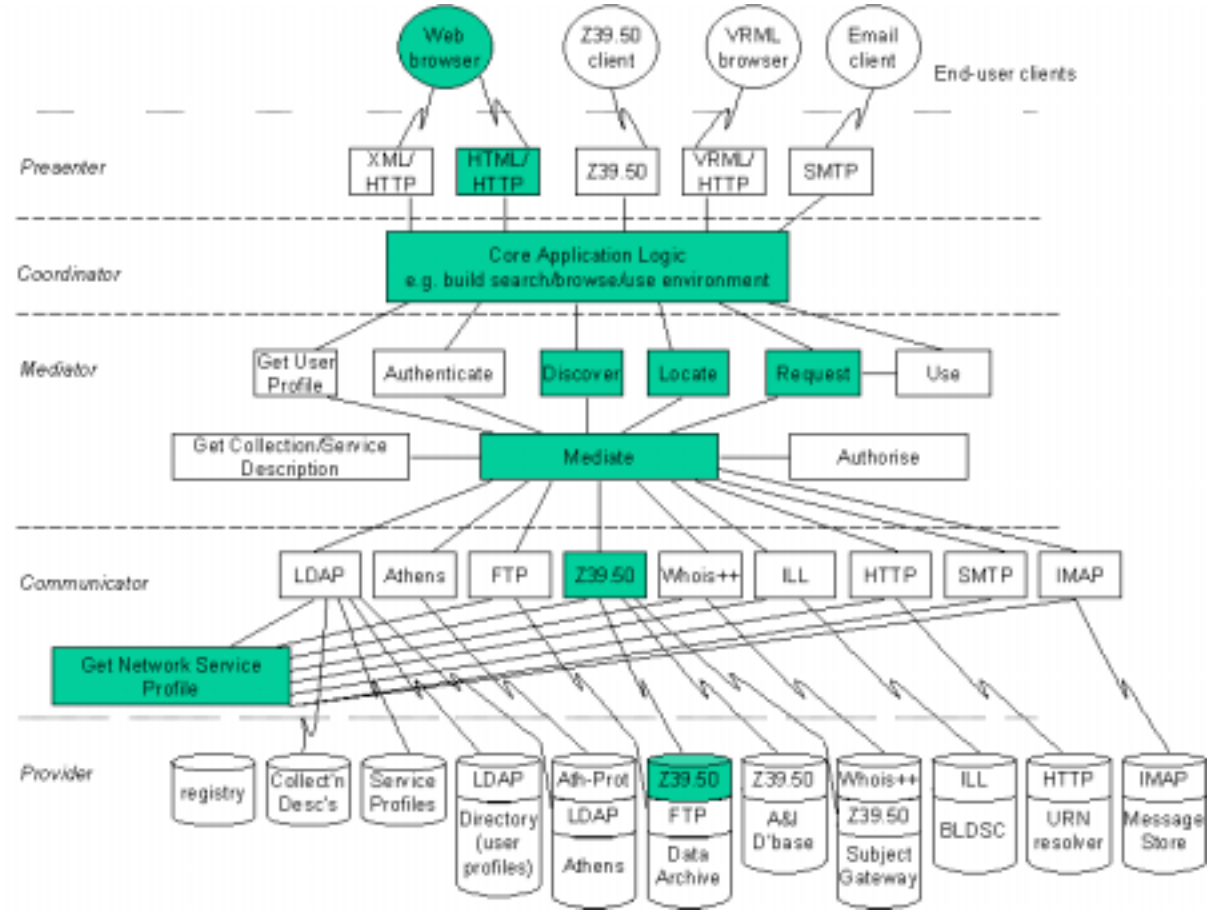


Figure 2.6: DEF related to MIA

The parts Presenter, Coordinator, Mediator,and Communicator are presently being developed. The first version will be ready 2000-07-01. All of these functions are implemented in one component, a Web to Z39.50 gateway called ZAP.

2.6.3.1 Presenter

The Web to Z39.50 gateway presents the user with HTML pages and forms generated on the fly from templates.

2.6.3.2 Coordinator

The HTML templates implement the coordinator functions.

2.6.3.3 Mediator

No real mediation takes place, apart from the conversion from HTML forms to valid Z39.50 queries and decoding of returned records into HTML structures.

2.6.3.4 Communicator

Based on internal tables reflecting the various Z39.50 profiles of the data providers this part provides Z39.50 access to databases.

2.6.3.5 Provider

Z39.50 databases. Each service provides two separate databases - one with metadata for resources and one with harvested full-text. All using the same basic metadata profile (with local additions).

2.6.4 Technical architecture

2.6.4.1 Standards

DEF is based on open standards. All portal services base their metadata definitions on Dublin Core. Furthermore they co-ordinate their definitions and maintain a core set of metadata fields. Each participating service can use local additions.

DEFcat data is based on MARC-records.

Involved profiles and metadata definitions:

- Vejviser metadata profile (in Danish)
- subject portal metadata profiles (in Danish)
- DanZig library catalogue Z39.50 profile
- Vejviser Z39.50 profile (in Danish)

2.6.4.2 Protocols

The DEF relies on HTTP and Z39.50.

2.6.4.3 Software

DEF Internet resource toolkit which includes:

- ZAP - a Z39.50 to Web gateway
- Z'Mbol - a Z39.50 database system
- Cataloging user interface
- Combine harvester

- Import/export utilities
- Common integrated configuration utility

2.6.4.4 Availability of software

The implementation is being done with a mixture of free and commercial software. Within the project a 'DEF Internet resource toolkit' is being implemented as a modular software based on separate components glued together with configuration files. Most components are/will be freely available.

2.6.5 References

Combine http://www.lub.lu.se/combine/

Danmarks Elektroniske Forskningsbibliotek (DEF): http://www.deflink.dk/english/

DanZig library catalogue Z39.50 profile: http://www.bs.dk/danzig/profil.htm

DEF vejviser: http://www.deff.dk/vejviser/

Subject portal metadata profiles (in Danish): http://dtv239.dtv.dk/fagportaler/metadataprofilerna.htm

Vejviser metadata profile (in Danish): http://dtv239.dtv.dk/format/formatbeskrivelse.htm

Vejviser Z39.50 profile (in Danish): http://dtv239.dtv.dk/lister/indexdata1.html

ZAP: http://www.indexdata.dk/pub/yaz/development/zap.tar.gz

Z'Mbol: http://www.indexdata.dk/zmbol/

2.7 Die Digitale Bibliothek Nordrhein-Westfalen (NRW)

Arthur N. Olsen, NetLab

2.7.1 Introduction

An integrated hybrid digital library system has been developed mainly for the academic sector in the German province of Nordrhein-Westfalen (North Rhine-Westphalia). Guest users can also access the system at the following URL: http://www.digibib-nrw.de/Digibib

2.7.1.1 Responsible agency

The period as a development project is now over and the responsible organisation is now HBZ (Online Utility and Service Center for Academic Libraries in North Rhine-Westphalia). Planning started in February 1998 with Bielefeld University Library as project co-ordinator. Two German companies - IHS technologies and Axion - have been responsible for software integration and development. The digital library system has been largely constructed utilising commercial software packages.

2.7.1.2 Description/Scope

Background information regarding the goals and systems design of the NRW system can be found in two documents dating from 1998:

- Konzept (Die Digitale Bibliothek NRW, 1998)
- Technishes Konzept (Groos, et al.,1998).

A brief overview of the project and technical description of this digital library system is presented in Habermann and Heidbrink (1999).

The functions that have been implemented are:

- Authentication via username and password
- Cross-searching of library catalogues, bibliographic databases, Internet resources, electronic periodicals and full text documents. Searching some systems is based on the Z39.50 protocol while in other cases unified searching of proprietary systems is enabled
- Integration with various document delivery systems for printed items
- Billing and payment procedures

The NRW system also includes support for access to electronic documents published by member organisations. The metadata describing the electronic documents is collected by a robot operated by NRW and loaded into the IR-system BRS after linguistic processing with the MILOS II system for enhanced retrieval.

2.7.1.3 Architectural diagram

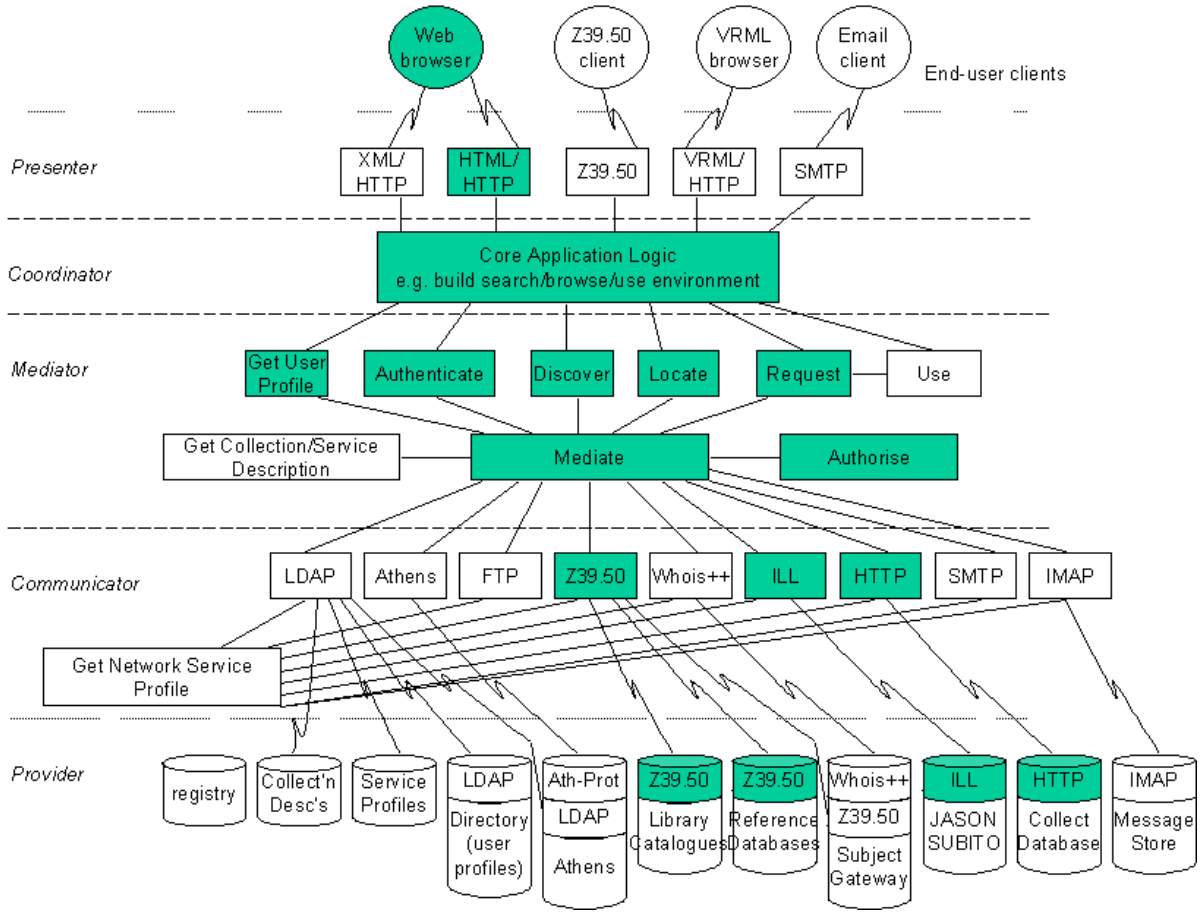


Figure 2.7: Die Digitale Bibliothek NRW related to MIA

2.7.2 Logical architecture (MIA framework)

2.7.2.1 Presenter

The user interface to the digital library is through standard browsers like Netscape and Internet Explorer

2.7.2.2 Coordinator

Authorisation, maintenance of user profiles and providing user services tailored to the individual user and site are provided in this layer

2.7.2.3 Mediator

Cross-searching and harmonisation of search result provided by the commercial products 'Query builder' and 'WebPAC' can be placed in this layer

2.7.2.4 Communicator

Access to some resources is based on the Z39.50 protocol. Link up with ILL services is also provided.

2.7.2.5 Provider

The Portal NRW brokers to a variety of library catalogues, databases and electronic documents both locally at the portal site and remote.

2.7.3 Technical architecture

2.7.3.1 Standards

The metadata for electronic documents conforms to the Dublin Core Metadata Element Set. The co-operating libraries have developed a profile for use in the NRW system (Sprick, 1999).

2.7.3.2 Protocols

The main protocols involved are HTTP and Z39.50.

2.7.3.3 Software

As mentioned above the NRW digital library system has been constructed using several software products from commercial companies:

- Query Server (Dataware)
- WebPAC (Epixtech)
- BRS/Search (Dataware)
- MILOS II software for verbal indexing based on automated linguistic processing and dictionaries

None of the software products mentioned are available under a General Public Licence. *Providing cross-searching capabilities for many heterogeneous European information gateways for the Renardus service could make it necessary to use commercial products such as Query Server.*

2.7.4 References

Digital Library Nordrhein-Westfalen: http://www.digilib-nrw.de/Digibib

IHS technologies: http://www.ihs.de/html/index.htm

Axion: http://www.axion-gmbh.de/

Dataware: http://www.dataware.com/technology/

Epixtech WebPAC: http://www.amlibs.com/product/htmlwebpac.htm

Die Digitale Bibliothek NRW, 1998, *Konzept*. Bielefeld: Bibliothek der Universität Bielefeld. http://www.ub.uni-bielefeld.de/digibib-nrw/konzept.htm

Groos, M., Hardt, J., Nold, A., Pieper D., Seiffert, F., Summann, F., 1998, *Die Digitale Bibliothek NRW - Technishes Konzept*. Bielefeld: Bibliothek der Universität Bielefeld. http://www.ub.uni-bielefeld.de/digibib-nrw/techkon.htm

Haberman, M., Heidbrink, S., 1999, Die Digitale Bibliothek NRW - Chronologie, Projektverlauf und Technische Beschreibung. *B.I.T. Online*, 2/1999. http://www.b-i-t-online.de/archiv/1999-02/nachrich/haberm/artikel.htm

Sprick, A., Tröger, B., Hoffmann, L., Hupfer, G., 1999, *Das Metadatenformat der Collect-Datenbank der Digitalen Bibliothek NRW*. Cologne: Hochschulbibliothekszentrum des Landes Nordrhein-Westfalen (HBZ). http://www.hbz-nrw.de/DigiBib/dokumente/allg/meta.html

2.8 ETB: the European Schools Treasury Broker

Arthur N. Olsen, NetLab

2.8.1 Introduction

2.8.1.1 Responsible agency

This is a major project co-ordinated by the European School network and funded by the Information Society Technologies program of the European Commission. The Swedish Ministry of Education / Committee for the European Schoolnet is co-ordinating partner. This new project builds on work done in the earlier project European Universal Classroom and the general collaboration and development work done under the auspices of the European Schoolnet.

2.8.1.2 Description/Scope

The project has just started so the following description is based on the project proposal (from Annex 1: Description of work):

Objective:

"The objective of the project will be to build a Web educational resource Metadata Networking and Quality Processing infrastructure for schools in Europe. This infrastructure aims to link together existing national repositories, encourage new publication, and provide a reliable level of quality and structure. There is no intention of reinventing national initiatives in this area, instead the objective is to add value to these systems while providing an interoperable layer to help teachers and students locate resources Europe wide. The proposal aims to build a simple yet effective distributed "Schoolnet Information Space". The material will be managed and organised according to defined rules in order to easily locate relevant resources. The educational user wants access to all data repositories, whatever indexing method is used or in whatever system they are supplied. The user, even in the world of decentralised, non-homogeneous data pools, demands from the system-developer to ensure that the following information requirements are met: The user should obtain only the most relevant documents, but if possible, also all of the most relevant ones according to his own information needs. This rich information space can form a reserve of educational material classified according to subject and resource type for use by teachers in preparing lessons, and by students for reference and research. It can contain guidelines and best practices for pedagogic reference and other material. An editorial interface aims to ensure a high quality of information and review. The project will enable and encourage trans-cultural and trans-national co-operation and communication and will enable individuals (students, teachers, administrators, parents) and workgroups to produce, handle, retrieve and communicate information in the languages of their choice, and to combine information resources from different regions and countries, and of different levels. It builds on existing or foreseen results of several projects: EUN MM1010, EUC and others. Deeper roots can be found in other projects like DESIRE, GEM and others".

Architecture Elements:

"The key to success will be the ease of use of tools for publishing resources to the information space and for locating relevant information in that space. It is paramount that the development take into account existing national systems and technical infrastructure available within the schools. It is therefore assumed that material is published at the schools or through national or regional repositories. It is the metadata describing the resources, and the multilingual subject classification and thesaurus that underlie the proposed European Treasury Browser system. At the technical level the planned components are as follows.

- A Web enabled multilingual educational subject classification and thesaurus to aid accessing and providing content.
- An intelligent data-entry system for the end-user including a metadata authoring tool with gateways to existing metadata systems, and a quality assurance procedure
- A dynamic metadata network to allow the flow of information across the Internet.
- A metadata registry with an intuitive search interface (client).

- Architecture elements are identified as follows:
- European Subject Thesaurus and Classification
- Intelligent Metadata Authoring Tools
- Metadata Network Transport.
- By harvesting Web sites (PULL)
- By searching repositories remotely (Z39.50)
- By transferring metadata records to the registry (PUSH)
- European Treasury Browser Metadata Registry"

As can be seen from the above material the goals and objectives of the ETB project are in many ways similar to those of Renardus. There is a focus on interoperability, metadata standards, multilingual access and harmonising access through controlled vocabularies. A major difference is the amount of development work that is planned regarding the construction of a new multi-lingual thesaurus. Development of a central ETB metadata registry and search client is envisioned as a first step followed by an expansion to a distributed registry system. Some architectural decisions seem to have been made already in the proposal, one of these is that there will be at least one union catalogue for metadata with the possibility for replication.

2.8.2 Logical architecture (MIA framework)

In the following the suggested architecture in the proposal will be related to the MODELS Information Architecture (MIA). The diagram bellow places functions from ETB into a framework developed by UKOLN in the document *An MIA view of DNER portals* (Powell, 1999). Relating the architectural elements of this project proposal to the more abstract MIA model is somewhat difficult. The components of ETB that can be related to this model are shaded.

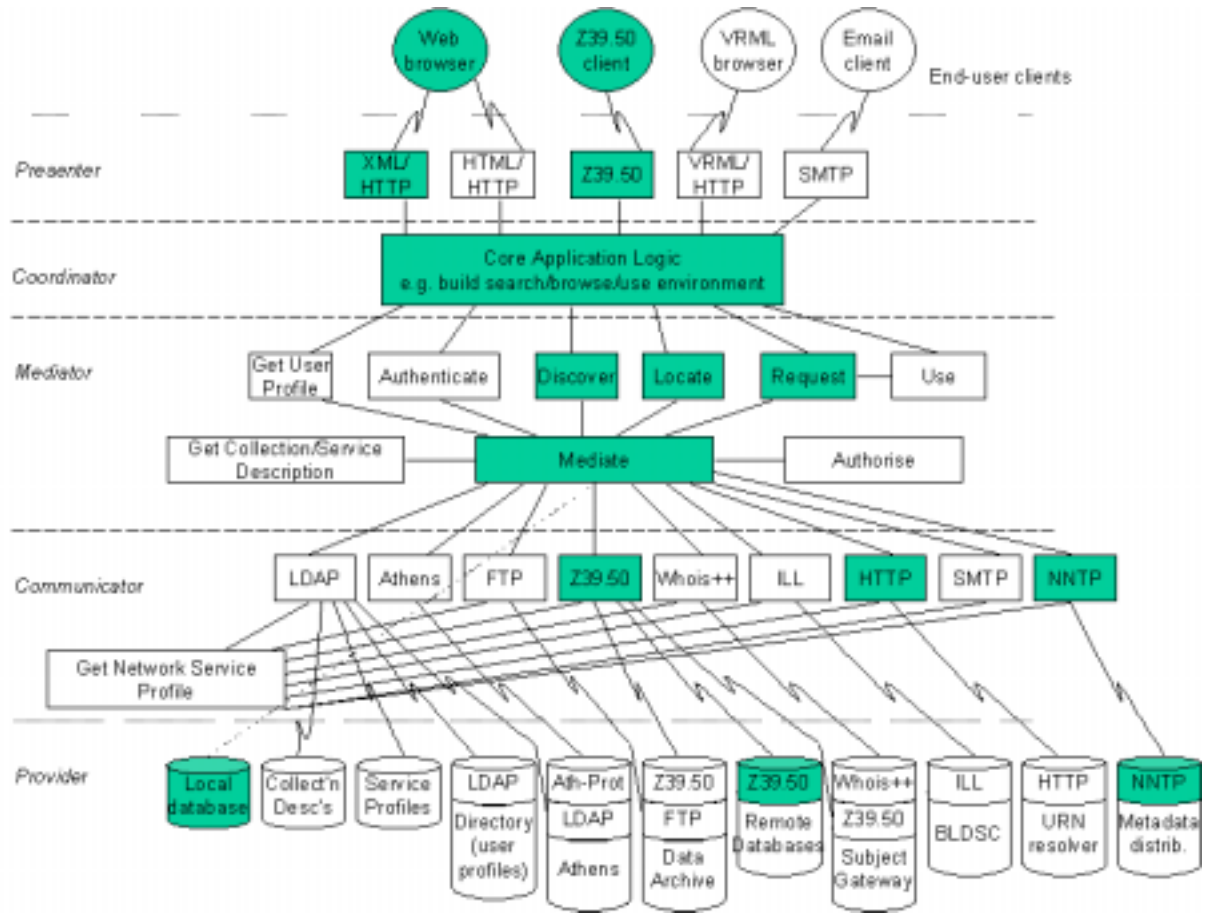


Figure 2.8: ETB related to MIA

2.8.2.1 Presenter

In the framework of ETB this will be purpose-built WWW routines for submission, thesaurus maintenance and searching supplemented by standard Z39.50 clients for searching

2.8.2.2 Coordinator

Most of the core functionality of the proposed system can be related to this layer:

- Metadata Registry Database
- Import of metadata into the database through harvesting, metadata authoring tools and network distribution
- Editorial control system
- Thesaurus module
- Searching services

2.8.2.3 Mediator

The need for complex mediation services is limited, it seems reasonable to relate the envisioned tools for Cross-lingual information retrieval to this layer

2.8.2.4 Communicator

The ETB proposal calls for using the Z39.50, NNTP and HTTP protocols that all can be related to this layer.

2.8.2.5 Provider

The services provided are limited to the local database and remote metadata registries within the realm of the ETB project

2.8.3 Technical architecture

2.8.3.1 Standards

The following standards are envisioned as part of the ETB project:

- Dublin Core metadata with extensions
- Metadata distributed with RDF/XML syntax
- The thesaurus that is part of the project will follow the relevant standards from ISO

2.8.3.2 Protocols

In addition to HTTP for access via WWW browsers Z39.50 is planned as a mode of access for searching. Metadata sharing, distribution and synchronisation is planned using the NNTP protocol.

2.8.3.3 Software

The choice of software for this project has not been made at this early date. Reuse, modification and packaging of existing software is planned. Some decisions regarding tools have been made. Gist, an information toolkit

developed by one of the patters of the ETB project will be utilised to develop much of the core functionality of the metadata repository. The Combine harvesting robot developed by NetLab will probably also be used. *This product which is in the public domain could be utilised for harvesting metadata in the Renardus context.*

2.8.4 References

European Universal Classroom: http://www.medianet.org/euc/

European Schoolnet: http://www.en.eun.org/front/actual

MODELS Information Architecture: http://www.ukoln.ac.uk/dlis/models/

Gist: http://gist.jrc.it/default/

Combine: http://www.lub.lu.se/combine/

2.9 EUropean Libraries and Electronic Resources in Mathematical Sciences (EULER)

Arthur N. Olsen, NetLab

2.9.1 Introduction

2.9.1.1 Responsible agency

EULER - EUropean Libraries and Electronic Resources in Mathematical Sciences - is a project within the realm of the Telematics for Libraries program of the European Union. The project started in April 1998 and is scheduled to continue until October 2000. Fachinformationszentrum (FiZ) Karlsruhe is the co-ordinator of the EULER project and two of the project partners - NetLab and the Niedersächsischen Staats- und Universitätsbibliothek (SUB) Göttingen - are also participants in Renardus.

The project has proceeded according to plans and a trial service is already accessible at the following URL: http://zaphod.lub.lu.se/euler/engine/engine.html

2.9.1.2 Description/Scope

The EULER service intends to offer a "one-stop shopping site" for users interested in mathematics by integrating bibliographic databases, library online public access catalogues, electronic journals from academic publishers, online archives of pre-prints and grey literature, and indexes of mathematical Internet resources. These will be made interoperable by using common Dublin Core based Metadata descriptions and a common user interface - the EULER Engine - will assist the user in searching for relevant topics in different sources in a single effort.

The focus of EULER is in many ways similar to Renardus, the differences being that EULER will broker to more diverse services in a specific subject realm. The service is based on a distributed architecture utilising Z39.50. The services brokered to represent library catalogues, subject bibliographies, electronic periodicals, pre-prints in full text and Internet resources. All the resources brokered to have been carefully harmonised by exporting and processing metadata to achieve consistency in regard to the common metadata format based on Dublin Core. Identical Z39.50 gateways have been installed at all sites.

As mentioned above the service is still in a test phase. Background and a detailed description of design considerations regarding the project are given in Berggren and Brümmer (1999) , the main project Web page contains reports for 1998 and 1999 and is located at : http://www.emis.de/projects/EULER/

2.9.1.3 Architectural diagram

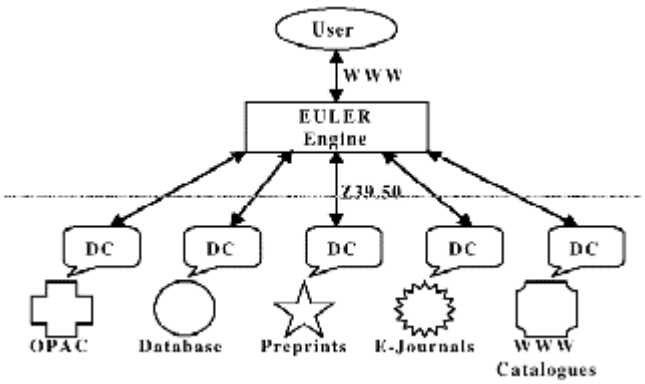


Table 2.9: EULER architecture

2.9.2 Logical architecture (MIA framework)

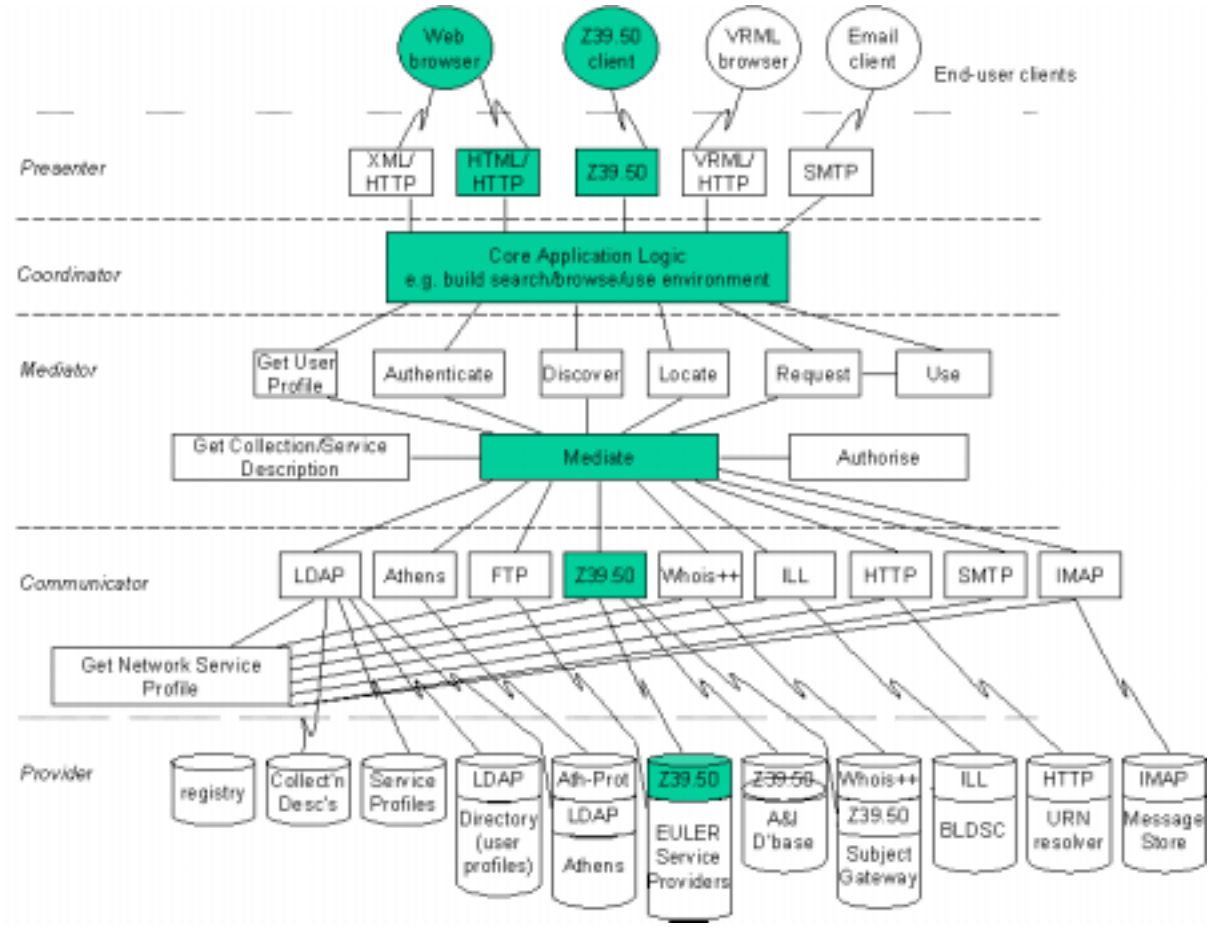


Figure 2.10: EULER related to MIA

2.9.2.1 Presenter

Access to the service is through standard Web browsers

2.9.2.2 Coordinator

The HTTP-Z39.50 gateway for cross-searching can be attributed to this layer, this software also contains functions for elimination of duplicates etc.

2.9.2.3 Mediator

Not relevant for EULER

2.9.2.4 Communicator

The system is based on the Z39.50 protocol

2.9.2.5 Provider

Currently EULER brokers the services of six different providers. All databases are distributed.

2.9.3 Technical architecture

2.9.3.1 Standards

The Dublin Core metadata element set is used, an EULER-DC profile has been developed

The syntax for entries and records is based on XML

The ISO-LATIN-1 character set is used

The Mathematics Subject Classification (MSC) is used for subject description

2.9.3.2 Protocols

As mentioned the Z39.50 protocol is central to EULER, an EULER specific profile has been developed

2.9.3.3 Software

All sofware used and developed in the EULER project is available free of charge for non-commercial use.

The HTTP-Z39.50 gateway is based on code from the EUROPAGATE project

The Z39.50 server and search system is based on the fielded free-text indexing and retrieval engine Zebra from Index Data.

As part of the project NetLab has developed a prototype of a metadata creation and editing tool. This tool is available at the following URL: http://euler.lub.lu.se/mdc/creator.cgi

The software used in the EULER project has a large amount of potential for use in the Renardus project, especially if an architecture based on distributed search using the Z39.50 protocol is chosen.

2.9.4 References

EULER Dublin Core Metadata Template: http://euler.lub.lu.se/mdc/creator.cgi

EULER pilot service: http://zaphod.lub.lu.se/euler/engine/engine.html

EULER project page: http://www.emis.de/projects/EULER/

EUROPAGATE project: http://europagate.dtv.dk/

Index Data: http://www.indexdata.dk/

Berggren, M., Brümmer, A., 1999, Design Considerations for the EULER project. Lund: NetLab. http://www.emis.de/projects/EULER/Reports/pD31.html

2.10 Finnish Virtual Library (FVL)

Risto Heikkinen, Jyväskylä University Library

2.10.1 Introduction

The Finnish Virtual Library (FVL) project - launched in 1995 and funded directly by the Finnish Ministry of Education - aims to form a foundation for a Finnish field-specific subject index of subject gateways. A collection of libraries have produced individual virtual libraries in 40 subject areas; these are now being converted into a gateway format, and offered as bilingual services in both Finnish and English.

2.10.1.1 Responsible agency

- Jyväskylä University Library (co-ordinator of the Finnish Virtual Library)
- Oulu University Library (administrator of the WHOIS++ server machine)

Other participants in Finnish Virtual Library are:

- Central Library of Theatre and Dance , Hanken's Library, Helsinki University of Technology Library, Helsinki School of Economics Library, Jyväskylä Polytechnic Library, Kuopio University Library, Library of the Finnish Literature Society, Library of Parliament, Library of Statistics, Sibelius Academy Library, Stakes - Information Service, Tampere University Library, University of Art and Design Library, Veterinary Medicine Library (Helsinki University), Viikki Science Library (Helsinki University)

2.10.1.2 Description/Scope

The FVL (Finnish Virtual Library) broker model is currently based on a ROADS-based WHOIS++ cross-searching service that brokers search access to 5 FVL gateways together with NOVAGate and the Swedish EELS gateway. More information on ROADS is available in the relevant broker review in this report (see: section 2.18). The 5 FVL gateways, which are cross-searched in Finnish Virtual Library, are:

1. Jyväskylä Virtual Library (subject fields: alcohol, drugs and other substance abuse, anthropology and folkloristics, applied linguistics, architecture, interior design and furniture design, education, Finnish art music, hydrobiology, literary research, psychology, sport science, theatre and dance): http://www.jyu.fi/library/virtuaalikirjasto/engroads.htm

2. Oulu University Virtual Library (subject fields: history of ideas, archaeology, space research, ecology, physics, geology, geriatrics, electronics and telecommunication technology, computers and data processing, chemistry - macromolecular structure research, geography, mathematics, Sami language and culture, Finnish history, applied mechanics, computer science): http://pc124152.oulu.fi/ROADS/haku.html

3. Tampere University Virtual Library (subject fields: information studies, journalism and mass communication, social sciences): http://virtuaalikirjasto.uta.fi/engroads.html

4. Virtual Library of Economics / Hanken's Library and Helsinki School of Economics Library (subject field: economics): http://helecon.hkkk.fi/virtuaalikirjasto/

5. Virtual Library Kuopio / Kuopio University Library (subject fields: clinical nutrition, environmental health, molecular medicine and gene therapy, neurosciences, nursing and health care, pharmacy): http://www.uku.fi/kirjasto/virtuaalikirjasto/

In the Finnish Virtual Library cross-searching does not cover the Elki gateway edited by the Library of Parliament and WebStat gateway edited by the Library of Statistics. The Elki is based on Trip Highway software. WebStat is based on ROADS software, so it is quite possible that it could be connected to FVL cross-searching in near future.

Cross-searching does not cover the following list-based field-specific virtual libraries either:

- Cultural studies

- New media
- Gerontology
- Energy technology
- Environmental protection technology
- Wood processing technology

Technically the broker utilises the WHOIS++ centroid functionality of ROADS software. WHOIS++ index server of Oulu University Library collects centroids once a day.

Almost all FVL gateways, which are brokered, utilise the same kind of modified ROADS/IAFA Document template. So there are not so many "mapping problems" between these gateways, whereas NOVAGate and EELS templates differ much more from our own template model. At present we don't get every field from these gateway templates in our search results.

In search results there is no information concerning the source database of the record.

The FVL launched its broker at the beginning of December 1999.

At this moment the broker has access to 6400 records (4100 records from 5 FVL gateways, 2300 records from NOVAGate and EELS)

Broker based cross-searching is an extensive but rough tool for information retrieval. Users can find this tool on the front page of FVL. More accurate search functions may be available on field-specific pages. Advanced search functions of cross-searching will probably be developed in the near future.

There have not been not so many technical problems in cross-searching. The system seems to be light-weight to use. The main problems, which have occurred, originate from the crashes of local subject gateway databases. Administrators of each subject gateway have to take care, that their databases are available for the broker all the time.

2.10.1.3 FVL Broker model and Renardus

It would be interesting to check out the possibilities to connect the FVL to Renardus service through our broker. Maybe that would be the easiest way for us to participate in Renardus. In our opinion the Renardus could offer connections to European gateway services both at broker and individual gateway level.

2.10.2 Logical architecture (MIA framework)

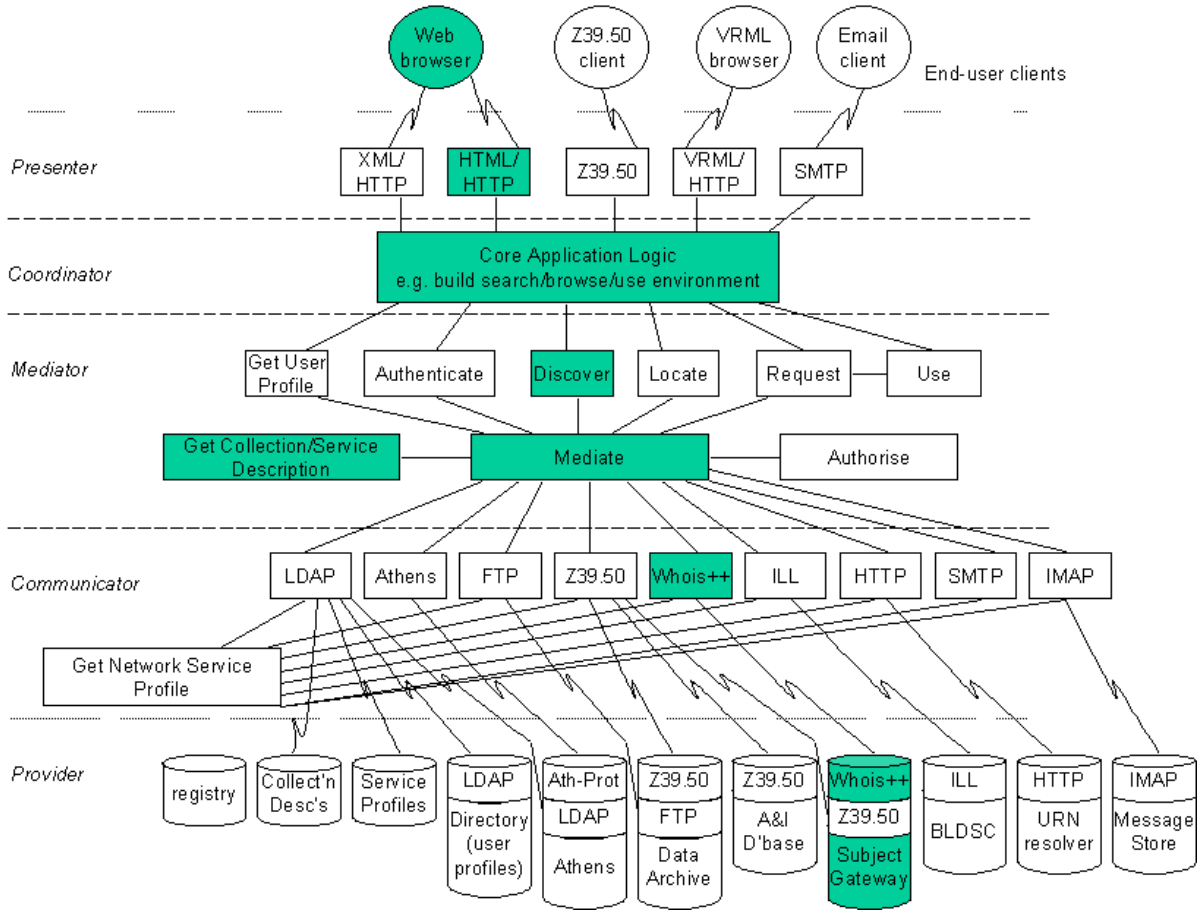


Figure 2.11: Finnish Virtual Library Broker Model related to MIA

2.10.3 Technical architecture

2.10.3.1 Standards

The FVL broker is based on modified ROADS/IAFA Document template. 4 FVL gateways, which are brokered, utilise this kind of template. The template has mapping possibilities to Dublin Core and Z39.50 Profiles.

2.10.3.2 Protocols

HTML/HTTP, WHOIS++, CIP (Common Indexing Protocol)

2.10.3.3 Software

ROADS (v2) software.

2.10.3.4 Availability of software

The FVL broker is based on public (and open source) ROADS software.

2.10.4 References

Finnish Virtual Library: http://www.jyu.fi/library/virtuaalikirjasto/engvirli.htm

NOVAGate: http://novagate.nova-university.org/

EELS: http://eels.lub.lu.se/

Elki gateway / Library of Parliament (subject fields: EU, law and justice, politics, public administration): http://www.eduskunta.fi/kirjasto/Elki/elkieng.html

WebStat / Library of Statistics (subject field: statistics): http://seitti.funet.fi:5000/etusivu_en.html

Day, M., 1999, *ROADS Interoperability Guidelines*. Bath: UKOLN, the UK Office for Library and Information Networking, 18 January. http://www.ukoln.ac.uk/metadata/roads/interoperability-guidelines/

Day, M., 1998, *The ROADS Template Registry, Template-Type: DOCUMENT*. Bath: UKOLN, the UK Office for Library and Information Networking, 10 June. http://www.ukoln.ac.uk/metadata/roads/templates/document.html

Heikkinen, R., 2000, FVL *ROADS-template and Example Record*, 15 March. http://www.sub.uni-goettingen.de/ssgfi/reynard/wp6/d6.1/meta_fvl.pdf [Renardus-password needed]

2.11 GAIA: Generic Architecture for Information Availability

Matthew J. Dovey, LAS

2.11.1 Introduction

2.11.1.1 Responsible agency

GAIA is an EU Funded project with the following partners:

Fretwell-Downing Data Systems Ltd (UK), Center for Tele-Information (Denmark), Codus Ltd (UK), Dialogue Communications Ltd (UK), IF Soft SRL (Romania), Ignis Technologies Ltd (Ireland), Index Data IS (Denmark), Kyros (Greece), National Centre for Popular Music (UK), Nexor Ltd (UK), North West Labs Ltd (Ireland), Q-Ray Documents BV (Netherlands), SeMeCo (Sweden), Technical Knowledge Center and Library of Denmark (Denmark), Teltec UCD-CS (Ireland), University of Sheffield (UK), University of the Aegean (Greece), University of Thessaloniki (Greece), Warp Records Ltd (UK).

2.11.1.2 Description/Scope

The GAIA architecture aims to provide a framework for multilateral information trading. It aims to provide an e-commerce solution to the location and delivery of information, content and delivery services, recognising the continued heterogeneity of information and that participant within a digital community should not be restricted to a single role.

The principal services offered by the GAIA model are:

- discovery of information, goods & services
- location of suppliers
- negotiation of service level in terms of quality, delivery & price
- delivery in the digital domain
- authentication, tariffing & payment management.

The GAIA architecture was tested within the GAIA Project in the domains of Music, Publishing, and Technical Data and over both Ethernet and cable television delivery mechanisms.

2.11.1.3 Architectural diagram

The basic functional diagram is given below – it is identical in essence with the standard MODELS overview. A customer would communicate with the GAIA broker using the paths outlined below. The broker would in turn communicate with the supplier using an identical model.

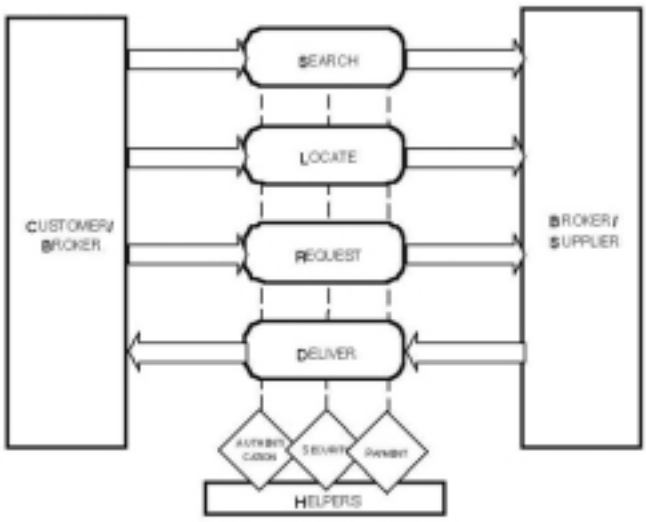


Figure 2.12: GAIA Architecture

To achieve this GAIA has Functional Units (FUs) and Functional Unit Managers (FUMs). The Functional Unit Managers perform a number of dedicated tasks co-ordinated by a brokering kernel acting as a communication bus. The Functional Unit Managers in the current GAIA architecture are:

- Search FUM – carries out a search for products that fit a particular user description returning a list of product identities
- Locate FUM – accepts product identifiers, discovers where they may be obtained and returns a list of suppliers and product locations
- Order FUM – manages negotiations between a customer and a supplier
- Item Delivery FUM – delivers file-structured items to the customer
- Streaming Delivery FUM – delivers real-time multimedia to the customer
- Customer FUM – provides an interface to the customer
- Alerting FUM – notifies customer about changes that may interest them
- Directory Services FUM – provides an interface between an external directory service and the brokering system
- Authenticate FUM – provides mechanisms to prove a users identity
- Payment FUM – provides a mechanism for payment from one actor to another

The Functional Unit communicates directly with the FUMs (but not the broker kernel) to carry out the work of the FUMs in a particular technology context (e.g. to use a particular protocol or standard), i.e. they behave as 'pluggable' "protocol adapters". A FUM may use several FUs.

The overall architecture is given below.

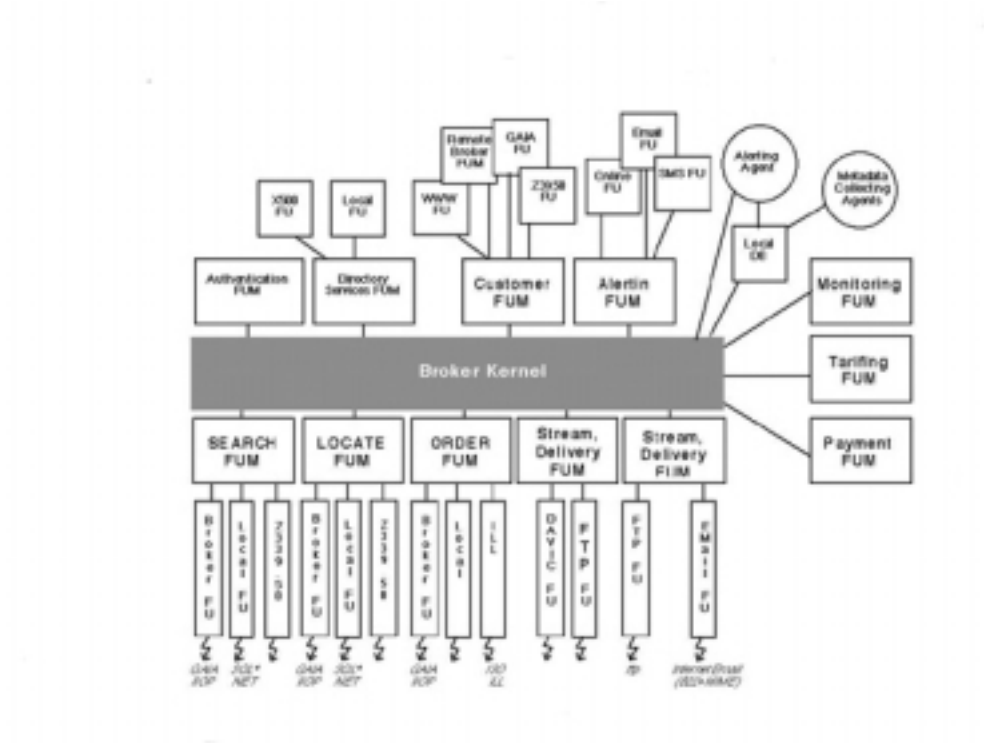


Figure 2.13: Overall GAIA Architecture

2.11.2 Logical architecture (MIA framework)

The GAIA Functional Units and Functional Unit Managers can roughly be mapped to the MIA framework as below. Important differences to note are that at the mediator level, the Functional Unit Managers and the broker kernel would be the other way around (i.e. the Functional Units at the communicator level would communicate to Functional Unit Managers not directly to the broker kernel as may be indicated in the diagram. Another difference is that both Functional Units and Functional Unit Managers are defined by their role in a given transaction, i.e. FUMs are not arbitrarily divided in the GAIA model into co-ordinators and communicators but may behave as either depending on the nature of the service they provide within a given transaction.

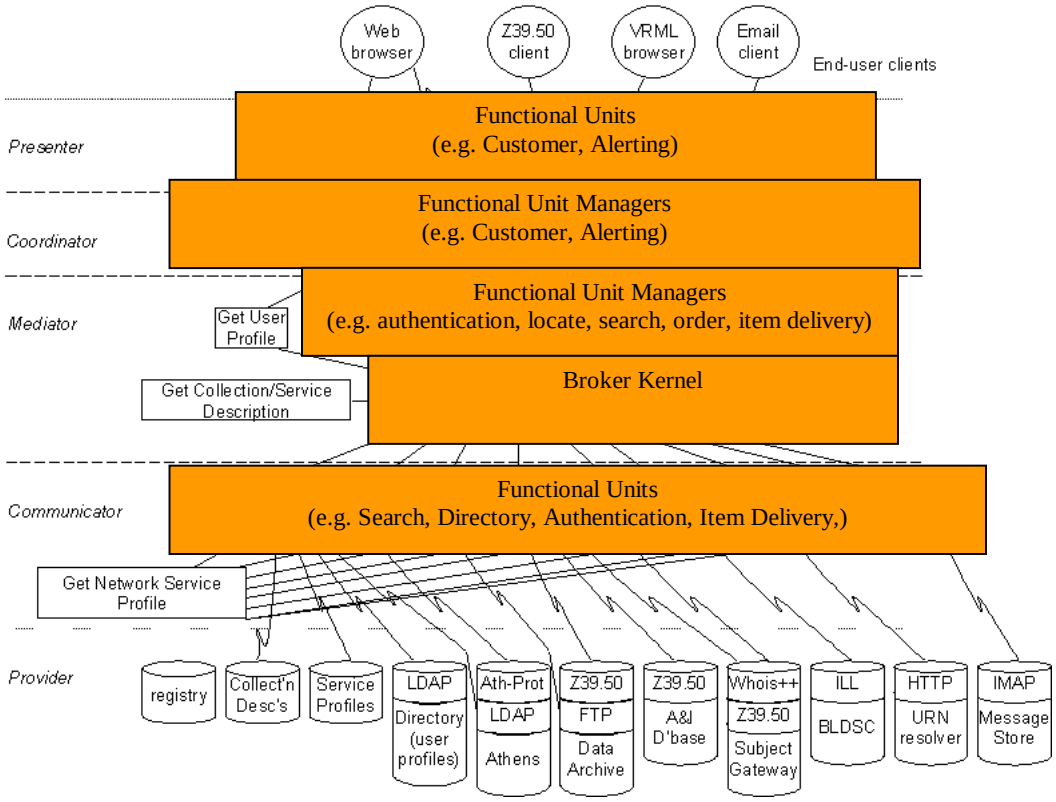


Figure 2.14: GAIA related to MIA

2.11.2.1 Presenter

The presenter is primarily provided by Functional Units that define protocols for use by the Customer Functional Unit Manager and the Alerting Functional Unit Manager. The GAIA implementation currently includes HTML, Z39.50, Short Message Service and e-mail. GAIA also includes FUs for another broker to interact directly with a GAIA broker and also for a dedicated GAIA Java client. Delivery FUs via the delivery FUMs would of course be involved in this layer as well as the communicator layers.

2.11.2.2 Coordinator

The Customer and Alerting FUMs provide for the co-ordinator layer, although the delivery FUMs would also be involved.

2.11.2.3 Mediator

The mediator layer would consist of the broker kernel which in GAIA would exist immediately below the Coordinator FUMs and below the kernel the additional FUMs (search, locate, delivery, authenticate etc.) would reside. Note that there are not specific internal items for obtaining user profiles or collection/service descriptions. These would be obtained via the authentication FUMs and directory services FUMs from external (in the sense that they are not intrinsic parts of the brokering layers) agencies.

2.11.2.4 Communicator

This layer is provided by various Functional Units for specific roles. FUs in the GAIA implementation include Z39.50, FTP, DAVIC and RTP (for stream delivery), ILL (for item ordering), LDAP, SET (for payment). Decision of appropriate FUs to use for a particular service (in MIA the role of the Get Network Service Profile) is handled by the FUM.

2.11.2.5 Provider

These would be the underlying supplier services. The Functional Unit abstraction from the business logic (inherent in the Functional Unit Managers) allows for heterogeneity of providers.

2.11.3 Technical architecture

2.11.3.1 Standards

The GAIA broker uses CORBA for communications and the interfaces are defined using Interface Definition Language (IDL). The project was active on a number of standard agencies including GEDI (Group on Electronic Data Interchange), SET (the standard system for secure credit card transactions designed and supported jointly by VISA and /MasterCard) and also prepared an IETF RFC which presents the GAIA Architecture for information of the Internet community in October 1998. They have also conformed to the IEC 61360 methodology for metadata.

2.11.3.2 Protocols

The current GAIA implementation uses the following protocols: HTML/HTTP, SMTP, SMS (Short Messaging Service), Z39.50, ISO ILL, DAVIC and RTP (for media streaming), LDAP, SET (for payment services) and SSLeay (for secure transactions).

2.11.3.3 Software

GAIA is developed primarily in Java and uses CORBA as its communications protocol. It has been tested on both VisiGenic and Orbix CORBA ORBs.

2.11.4 References

GAIA Project: http://www.syspace.co.uk/GAIA/

Hands, J., 1998, GAIA: Generic Architecture for Information Availability. In: InfoWin Thematic Issues: *Information Brokerage*. Berlin: Deutsche Telekom Berkom GmbH, January. http://www.infowin.org/ACTS/ANALYSYS/PRODUCTS/THEMATIC/BROKERAGE/gaia.html

Koutsabasis, P., Darzentas, J.S., Spyrou, T., Darzentas, J., 1998a, An interaction agent in a brokerage environment: methodology, design and application. *Proceedings of first international workshop on interaction agents, L' Aquilla, Italy, 24 May 1998*, pp. 58-61.

Koutsabasis, P., Darzentas, J.S., Spyrou, T., Darzentas, J., 1998b, Intelligent agents and information brokerage. *Proceedings of 12th National Conference of the Greek Operational Research Society, Samos, Greece, 4-6 September 1998* (in press).

Koutsabasis, P., Darzentas, J.S., Spyrou, T., Darzentas, J., 1999a, Facilitating User–System Interaction: the GAIA Interaction Agent. To appear in: *Proceedings of 32nd Hawaiian International Conference on System Sciences (HICSS 99), Hawaii, 5-8 January 1999* (in press).

Koutsabasis, P., Darzentas, J.S., Spyrou, T., Darzentas, J., 1999b, GAIA Interaction Agent: proactive assistance to users of electronic brokerage systems. To be published in: *Computer Communications Journal* (Elsevier) and/or *Computer Networks and ISDN Systems* (as part of the CLIMATE initiative for a special issue on Brokerage and Agents), July - September 1999.

Langer, T., Adametz, H., Fellien, A., 1999, *Feasibility Study: MALVINE Information Brokerage Platforms.* MALVINE project document, May, pp. 21, 58-60. http://www.malvine.org/malvine/eng/publications.html

2.12 Harvest

Martin Hamilton, LUT

2.12.1 Introduction

2.12.1.1 Responsible Agency

Harvest was originally developed by the Internet Research Task Force Research Group on Resource Discovery (IRTF-RD) as part of an ARPA-funded project (Bowman, et al., 1994). The final official Harvest release (1.4pl2) forms the basis for current open source development.

2.12.1.2 Description/Scope

The Harvest system was originally intended to be a generalised resource discovery system, with support for replication (mirroring) and object caching. In the event, replication never fully materialised, and the object caching component was so successful that it was eventually separated out into a package in its own right - Squid, formerly the Harvest Object Cache.

This leaves the remaining components of Harvest - the Broker and the Gatherer. The goal of the Gatherer is to perform automated (robot based) indexing of arbitrary content, using the Essence system from the University of Arizona (included in the package). This utilises a plug-in architecture to support any file format that can have a 'summariser' program written for it.

Summarisers produce Summary Object Interchange Format (SOIF) metadata, which the Gatherer makes available via a daemon (gatherd) using the Harvest Gatherer protocol. Summariser programs are available for a variety of file formats, including HTML. The role of the Broker is to periodically fetch SOIF objects from one or more Gatherers and index them. The Broker itself may then be searched using the Harvest Broker protocol. As with the Gatherer summarising process, alternative search engines can be plugged into the Broker - though this is a more complex process and requires C programming skills. The glimpse and swish full text search engines (amongst others) are supported.

The only mode of interaction with the system available to the end user is to perform a search via a Web form, which will cause a set of search results to be returned. Individual search results can be selected for display. Some very limited Web based admin features are provided - e.g. one can initiate the gathering operation.

2.12.2 Logical architecture (MIA framework)

2.12.2.1 Presenter

Perl CGI script (nph-search) which speaks the Harvest Broker protocol to the Broker and returns HTML to the end user

2.12.2.2 Coordinator - Mediator - Communicator - Provider

The Harvest Broker serves in this role

The Gather component of Harvest effectively constitutes an additional layer that MIA does not directly address.

2.12.3 Further details

2.12.3.1 Software

Version Reviewed: 1.6.1

Download: http://www.tardis.ed.ac.uk/harvest/ or: ftp://ftp.tardis.ed.ac.uk/harvest

Status: Open source software distributed under the terms of the GNU General Public License.

Support: Community support available via the harvest-develop@tardis.ed.ac.uk mailing list and the comp.infosystems.harvest Usenet newsgroup.

Protocols: User interacts via HTTP. Behind the scenes, a dedicated Harvest Broker protocol provides for search and retrieval, and a dedicated Harvest Gatherer protocol provides for bulk transfer of metadata from Gatherers to Brokers.

Formats: Only the SOIF metadata format (originally developed for Harvest) is supported.

Platforms: Any modern Unix-like system.

Prerequisites: HTTP server, Perl, C compiler.

2.12.3.2 Decision points

The Harvest developers have taken great care to design an open-ended system using network protocols, programming APIs and a standard metadata format (SOIF). These make it possible for them to abstract away the implementation details underlying the search engine. However, whilst Harvest as a package goes to great lengths to implement this modular design, it still only provides a minimal set of features. The availability of a wide range of open source software in this area meant that it was not necessary for the Harvest developers to build:

- A search engine - they simply use existing indexers such as glimpse.
- A system for summarising content - this is what essence does.
- An HTTP server for the end user visible search interface - a large number of servers implementing CGI are available.

Thus, what we call "Harvest" is essentially the plumbing to connect these components together over a network.

2.12.4 Conclusions

Although Harvest would appear from this description to be a well-structured package, it has a number of problems:

- Suffers from code bloat - in that the source code for a number of external packages is included in the distribution. This situation has been improved, though, as earlier versions of Harvest included even more external software.
- Uses several programming languages (C, Perl and shell scripts) for the same tasks - without code re-use in many cases.
- Most of its features are un(der)developed, e.g. Web based admin.
- The Harvest protocols and the SOIF metadata format are very poorly documented. This is unfortunate, given the Harvest developers' stated aim of encouraging commercial (re)implementations and standardisation.

These are being rectified in an ongoing effort to develop a next generation version of Harvest, which is being written entirely in Perl. In the meantime, the Harvest 1.x series is still being maintained.

Although Harvest is no longer widely used, there is still no commonly accepted dedicated search and retrieval protocol for the Internet, and it can be argued that the Harvest Broker protocol is still a contender for this role.

The Harvest Gatherer protocol is one of a very small number of purpose-built bulk metadata transfer protocols, although it has effectively been superseded by the Common Indexing Protocol. The SOIF metadata format has been used in several other contexts but has effectively been superseded by Dublin Core, XML and RDF - though the shape of the eventual metadata format that will come from these has yet to become clear.

Building systems that support the Harvest protocols and the SOIF metadata format means that they will they will be interoperable with Harvest, but will confer little or no additional benefit. This said, SOIF and the Harvest protocols are simple to implement and deploy, which means that they can be provided at a very low cost, and interoperability with existing resource discovery systems such as Harvest can be expected to increase take-up for a new system.

2.12.5 References

Harvest: http://www.tardis.ed.ac.uk/harvest/ or ftp://ftp.tardis.ed.ac.uk/harvest

Bowman, C.M., Danzig, P.B., Hardy, D.R., Manber, U., Schwartz, M.F., 1994a, *Harvest: a scalable, customizable discovery and access system.* Technical Report CU-CS-732-94. Boulder, Colo.: University of Colorado at Boulder, Department of Computer Science, July.

2.13 ht://Dig

Martin Hamilton, LUT

2.13.1 Introduction

2.13.1.1 Responsible agency

ht://Dig was originally written by Andrew Scherpbier whilst working at San Diego State University, but is currently being developed and maintained by a large group of volunteers as a community led project.

2.13.1.2 Description/Scope

ht://Dig is a Web based indexing and searching package, primarily written in C++. It consists primarily of two components - a robot based indexing (or "digger") program *htdig*, and a searching program *htsearch*. The *htdig* program performas a recursive traversal of a nominated Web site and stores its results in a custom database format. The *htsearch* program is intended to run under CGI, and performs a search of the database using the CGI parameters passed to it on invocation. The result will be an HTML document containing the result set.

2.13.1.3 Architectural diagram

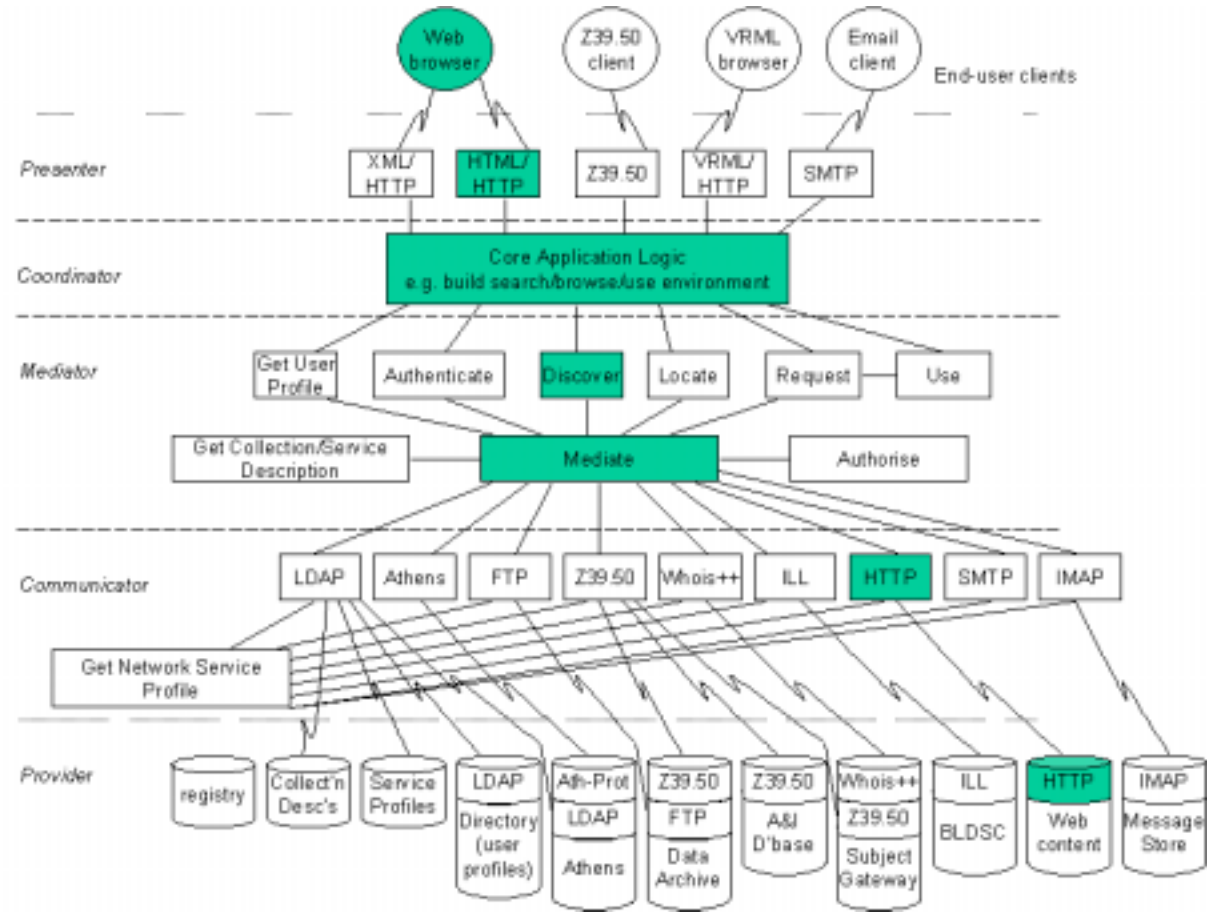


Figure 2.14: ht://Dig architecture related to MIA

2.13.2 Logical architecture (MIA framework)

2.13.2.1 Presenter - Coordinator - Mediator - Communicator- Provider

C++ CGI program (*htsearch*) which interacts with the end user and accesses the database directly.

The digging component of ht://Dig effectively constitutes an additional layer that MIA does not directly address.

2.13.3 Technical architecture

2.13.3.1 Standards

The end user interacts via HTTP with a CGI program *htsearch* which provides the actual search capabilities. The database being searched is constructed using another program, *htdig*, which performs HTTP based traversal and indexing starting at a nominated URL. The ht://Dig database is not accessible via any dedicated search and retrieval protocol.

2.13.3.2 Protocols

There is no native ht://Dig metadata format (the index is created directly by *htdig*), and no metadata interchange formats are supported. Metadata embedded in HTML documents will be processed by ht://Dig, however.

2.13.3.3 Software

Version Reviewed: 3.2.0b1 (4th February 2000)

Download: http://www.htdig.org/

Status: Open source software distributed under the terms of the GNU General Public License.

Support: Community support is available via the *htdig@htdig.org* mailing list.

Platforms: Any modern Unix-like system. Development and testing primarily done using Linux.

Prerequisites: HTTP server, Perl, C/C++ compiler

No additional software is required to get a server up and running with ht://Dig.

2.13.4 Conclusions

ht://Dig is optimised for rapid indexing and searching, and as a consequence does not support most of the modular functionality found in some of the other packages we review. Although ht://Dig is a resource discovery system, without modification it would not be useful in the role of subject gateway - since it is designed solely for full text indexing of Web pages. This said, there may be some lessons here for the designers of resource discovery toolkits which are expected to be used to host extremely popular Websites - e.g. that volume testing should be carried out to simulate the effect of a "success disaster" such as having your Website mentioned prominently in the news media.

It is possible to supply plug-in summariser programs for indexing arbitrary types of content using ht://Dig, in a similar manner to Harvest's use of the Essence summariser. This feature of ht://Dig is, however, aimed at primarily at the extraction of full text from the likes of PDF (Adobe Acrobat) and Microsoft Word documents.

2.14 Isaac Network

Martin Hamilton, LUT

2.14.1 Introduction

2.14.1.1 Responsible agency

The Isaac Network was an initiative of the Internet Scout Project (funded by the US National Science Foundation) at the Department of Computer Science at the University of Wisconsin-Madison. Although the Isaac software was never formally released, a number of architecture overview papers were published - these are used as the basis for this review.

2.14.1.2 Description/Scope

The Issac Network software is conceptually very similar to ROADS, but provides a much more limited set of functionality - restricted to search and retrieval, although cross-searching of multiple Isaac nodes is supported. There is no equivalent to the Web based server admin and maintenance tools provided with ROADS. It is assumed that records will be created by human cataloguers - possibly with automated assistance, and there is no direct support for robot based harvesting of metadata.

Isaac supports three distinct services:

- Metadata repository - provides storage for metadata records, which may be fetched by other Isaac nodes.
- Index service - gathers metadata from repositories, indexes it, and makes it available for searching.
- Search service - provides a Web based front end for searching multiple index services.

The metadata repository is also responsible for generating Common Indexing Protocol Tagged Index Objects (TIOs) of the data that it holds.

2.14.2 Logical Architecture (MIA Framework)

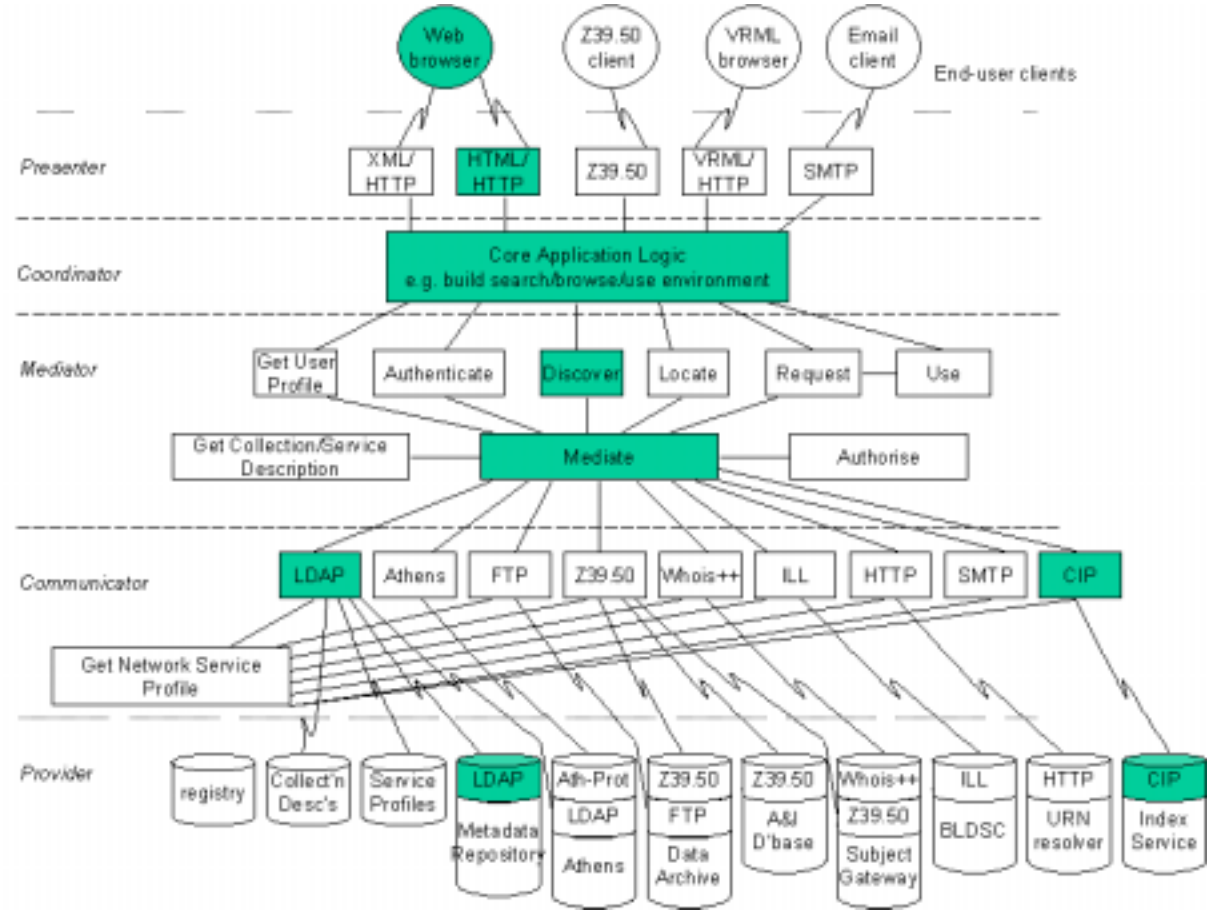


Figure 2.15: ISAAC related to MIA

2.14.2.1 Presenter - Coordinator - Mediator - Communicator

CGI search script acting as LDAP client processing referrals and searching multiple LDAP servers

2.14.2.2 Provider

LDAP server acting as metadata repository

2.14.3 Technical architecture

2.14.3.1 Standards

Metadata records in Isaac may be interchanged using the LDAP Interchange Format (LDIF). Internally the LDAP schema used is the Dublin Core LDAP/X.500 draft proposed by Martin Hamilton in 1996.

2.14.3.2 Protocols

Isaac uses LDAP rather than WHOIS++ as its search and retrieval protocol. Like ROADS, the Common Indexing Protocol is used to provide distributed indexing and searching, but the Isaac software does not support WHOIS++ centroids (RFC 1913 and RFC 1914).

2.14.3.3 Software

Status: Not publicly available.

Platforms: Several modern Unix-like systems, including Solaris, FreeBSD and Linux.

Prerequisites: HTTP server for search service, possibly others.

No additional software is mentioned as a requirement in the Isaac papers, although they are vague about the technical details of the Isaac software.

As with ASF, the core functionality of the Isaac Network software is provided by already written third party software - in this case the University of Michigan LDAP distribution. The Isaac developers have extended this to support some of the LDAP version 3 functionality and implemented the Common Indexing Protocol separately using Perl.

2.14.4 References

Lukas, C., Roszkowski, M., 1999, *The Isaac Network: LDAP and Distributed Metadata for Resource Discovery*. Paper presented at the Third IEEE Meta-Data Conference, Bethesda, Md, 6-7 April 1999. http://computer.org/proceedings/meta/1999/papers/46/clukas.html

Roszkowski, M., Lukas, C., 1998, A Distributed Architecture for Resource Discovery Using Metadata. *D-Lib Magazine*, September. http://www.dlib.org/dlib/june98/scout/06roszkowski.html

2.15 Jointly Administered Knowledge Environment (jake)

Martin Hamilton, LUT

2.15.1 Introduction

2.15.1.1 Responsible agency

The Jointly Administered Knowledge Environment (jake) is a community driven project, with no overall co-ordinator. Development is done by volunteers and co-ordinated using mailing lists.

2.15.1.2 Description/Scope

Unlike most of the other systems we review here, Jake is both a software development and database gathering project. It aims to provide both a database (in SQL format) of electronic resources (such as online journals) and also software to provide searching and cross-referencing capabilities.

The latter is currently done using four PHP3 scripts, which are still in fairly early stages of development.

2.15.2 Logical architecture (MIA framework)

2.15.2.1 Presenter - Coordinator - Mediator - Communicator

The user interacts with a PHP3 script which uses the PHP3 MySQL module to search jake.

2.15.2.2 Provider

The jake MySQL server

2.15.3 Technical architecture

2.15.3.1 Standards

Jake is built on top of SQL, with the real result of the project to date being a database of some 21,000 records which is supplied in SQL format suitable for loading in MySQL or another compatible SQL implementation.

As has been noted, the sample jake search scripts which have been developed to date depend on PHP3.

2.15.3.2 Protocols

Jake simply uses HTTP to interact with the end user.

2.15.3.3 Software

Version Reviewed 0.5 (of the jake-misc package - 2nd March 2000)

Download: http://jake.med.yale.edu

Status: Open source software and database distributed under the terms of the GNU General Public License.

Support: Community support is available via the *jake-list@vectra.med.yale.edu, jake-devel-list@vectra.med.yale.edu, jake-updates-list@vectra.med.yale.edu* mailing lists.

Platforms: Any modern Unix-like system.

Prerequisites: Software designed for use with Apache using the PHP3 module and the MySQL database. May work with other permutations of SQL server.

Although PHP3 is commonly shipped with modern (free) Unix distributions, e.g. RedHat Linux, MySQL is less frequently shipped - presumably due to its licensing conditions. Since no special MySQL features are relied upon, it should be possible to use the jake database with other SQL servers - e.g. the Postgres SQL server that is more commonly shipped.

2.15.4 Conclusions

Jake is very interesting for the developers of a new resource discovery toolkit, since it does not support any of the `traditional' mechanisms used in Internet resource discovery systems - instead the jake developers have chosen to concentrate on building their database and the associated data model. This does not preclude the later addition of features such as a search and retrieval protocol, metadata interchange or distributed indexing.

Even without these features jake constitutes a very powerful and useful system - and also a very high performance one, since the underlying MySQL database has been optimised for high volume applications such as dynamic Web site hosting. Given the lack of consensus on Internet search and retrieval protocols, it is arguably a good thing that jake is not tied to a particular protocol or associated architecture.

2.16 Networked Computer Science Technical Research Library (NCSTRL/Dienst)

Arthur N. Olsen, NetLab

2.16.1 Introduction

NCSTRL - the Networked Computer Science Technical Research Library is a distributed digital library in the field of Computer Science. The NCSTRL network has been expanded to Europe in co-operation with ERCIM and includes documents from over 150 participating institutions and archives. Links have also been made to arXiv - the Los Alamos e-Print archive. The technical foundation of NCSTRL is Dienst - a protocol and architecture for digital libraries.

2.16.1.1 Responsible agency

The NCSTRL digital library and the Dienst protocol have been developed with support from the US Advanced Research Projects Agency (ARPA) and other sources. Development has to a large degree been done at Cornell University and Xerox (Davis and Lagoze, 1999)

2.16.1.2 Description/Scope

The design philosophy that has motivated NCSTRL/Dienst is based on an open object oriented approach. The service can be divided into the following components:

- A *Repository Service* - stores digital documents (according to a defined document model). Each of these has a unique name and may exist in multiple versions, each with different components and formats. Mapping from unique name to repository is done by an external naming service.
- An *Index Service* accepts queries and returns lists of document identifiers matching those queries.
- A *Query Mediator Service* dispatches queries to appropriate index servers.
- An *Info Service* returns information about the state of a server hosting one or more services.
- A *Collection Service* provides information on how a set of services interacts to form a logical collection.
- A *Registry Service* stores information about (human) users of services of a collection.

The Dienst *User Interface Service* provides interaction with the system and is adapted to using normal Web browsers.

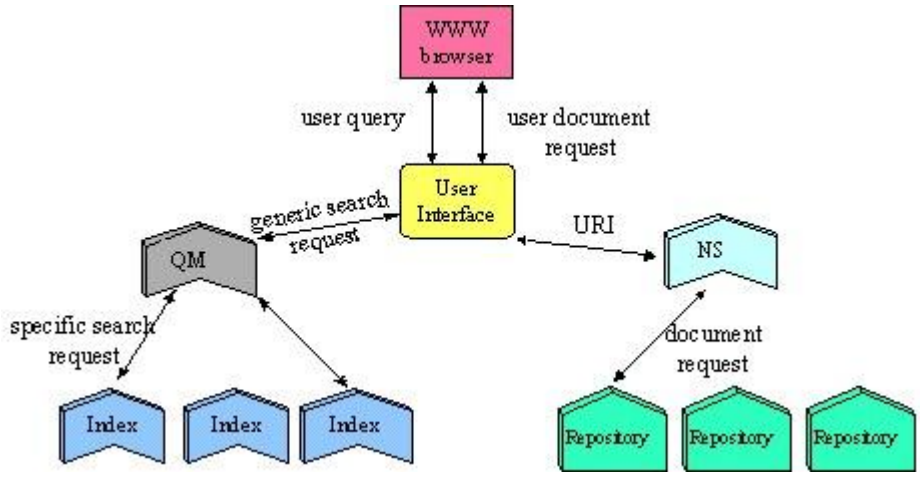
Dienst also relies on a *name service* that provides unique identifiers for digital documents based on CNRI's Handle system.

The Dienst protocol has been revised several times to reflect changes in the services available and the growth of the digital library. A new development is a subset of the protocol that has been designed as part of the Open Archives Initiative. This initiative builds on work regarding NCSTRL and at Los Alamos to develop a larger degree of interoperabilty between electronic archives. The Open Archives Initiative provides an important foundation for developing better digital library services to end-users. The subset of the Dienst protocol allows third parties to develop advanced new services based on harvesting metadata from open archives that adhere to well documented standards (Van de Sompel, 2000). An Open Archives Metadata set has also been developed.

The development of NCSTRL has shown that a distributed digital library service built on open standards can provide a high quality service for end-users.

2.16.1.3 Architectural diagram

The following diagram (from the Dienst Architecture summary description) shows an overall conceptual overview of the different components of Dienst:



NS = Name service

QM = Query Mediator

Figure 2.16: Dienst Architecture

2.16.2 Logical architecture (MIA framework)

The Dienst Architecture is based on a number of clearly defined services that when combined constitute a digital library service. Some of the services in Dienst can be directly related to layers in the MIA model. Other services are not so easily placed.

2.16.2.1 Presenter

This layer of the MIA model corresponds to the *user interface* service in Dienst.

2.16.2.2 Coordinator

Query moderation and the *collection* service.

2.16.2.3 Mediator

The *registry* service regarding users.

2.16.2.4 Communicator

The *info and name* services.

2.16.2.5 Provider

The *repository and index* services fit neatly into the provider layer.

2.16.3 Technical architecture

2.16.3.1 Standards

URN's based on the handle system

XML as the external format for metadata

Dienst builds upon results from an earlier - DARPA-funded - project that concerned the electronic distribution of computer science reports: the CS-TR (Computer Science Technical Reports) Project. A standardised format for the exchange of bibliographic records was developed that has some resemblance to the Dublin Core metadata set (RFC 1357, now RFC 1807). Metadata from NCSTRL is now available following a new standard called the Open Archives Metadata Set. This is a minimal format based on Dublin Core.

2.16.3.2 Protocols

The Dienst protocol is the cornerstone of the NCSTRL system. HTTP is used as a transport for Dienst.

2.16.3.3 Software

The software developed for implementing digital libraries based on Dienst is copyrighted but freely available. In addition to the Dienst software itself a complete installation requires a number of other software packages including Perl and Apache. Indexing services in Dienst are configured for the freeWAIS-sf search engine.

NCSTRL has shown that a distributed digital library based on Dienst can provide a scalable and efficient solution. *Using Dienst as a foundation for a Renardus service is a distinct possibility.* This would require participants to install Dienst services in addition to there own systems.

2.16.4 References

NCSTRL home page: http://www.ncstrl.org/

arXiv the Los Alamos e-Print archive: http://arXiv.org/

Dienst architecture: http://www.cs.cornell.edu/cdlrg/dienst/architecture/architecture.htm

Dienst overview: http://www.cs.cornell.edu/cdlrg/dienst/DienstOverview.htm

Dienst software: http://www.cs.cornell.edu/cdlrg/dienst/software/DienstSoftware.htm

freeWAIS-sf: http://ls6.cs.uni-dortmund.de/ir/projects/freeWAIS-sf/old/

Handle system: http://www.handle.net/

Open Archives Initiative: http://www.openarchives.org/

Open Archives Metadata Set: http://www.openarchives.org/sfc/sfc_oams.htm

Open archives subset of Dienst protocol: http://www.cs.cornell.edu/cdlrg/dienst/protocols/OpenArchivesDienst.htm

RFC 1807: Lasher, R., Cohen, D., 1995, *A format for bibliographic records*. http://www.ietf.org/rfc/rfc1807.txt

RFC 1807. In: Dempsey, L., Heery, R., 1997, *A review of metadata: a survey of current resource description formats*. DESIRE project Deliverable 3.2 (1). http://www.ukoln.ac.uk/metadata/desire/overview/rev_19.htm

Davis, J.R., Lagoze, C. 2000, NCSTRL: design and deployment of a globally distributed digital library. *Journal of the American Society for Information Science*, 51 (3), pp. 273-280.

Van de Sompel, H., Lagoze, C., 2000, The Santa Fe Convention of the Open Archives Initiative. *D-lib Magazine*, 6 (2) February. http://www.dlib.org/dlib/february00/vandesompel-oai/02vandesompel-oai.html

2.17 Resource Discovery Network: Resource Finder

Andy Powell, UKOLN

2.17.1 Introduction

2.17.1.1 Responsible agency

The Resource Discovery Network Centre (RDNC), jointly run by staff at UKOLN and King's College, London.

2.17.1.2 Description/Scope

The RDN ResourceFinder service is a ROADS-based WHOIS++ cross-searching service that brokers search access to the SOSIG, Biz/ED, EEVL and OMNI RDN Internet Resource Catalogues. More information on ROADS is available in the relevant broker review in this report (see: section 2.18).

2.17.1.3 Architectural diagram

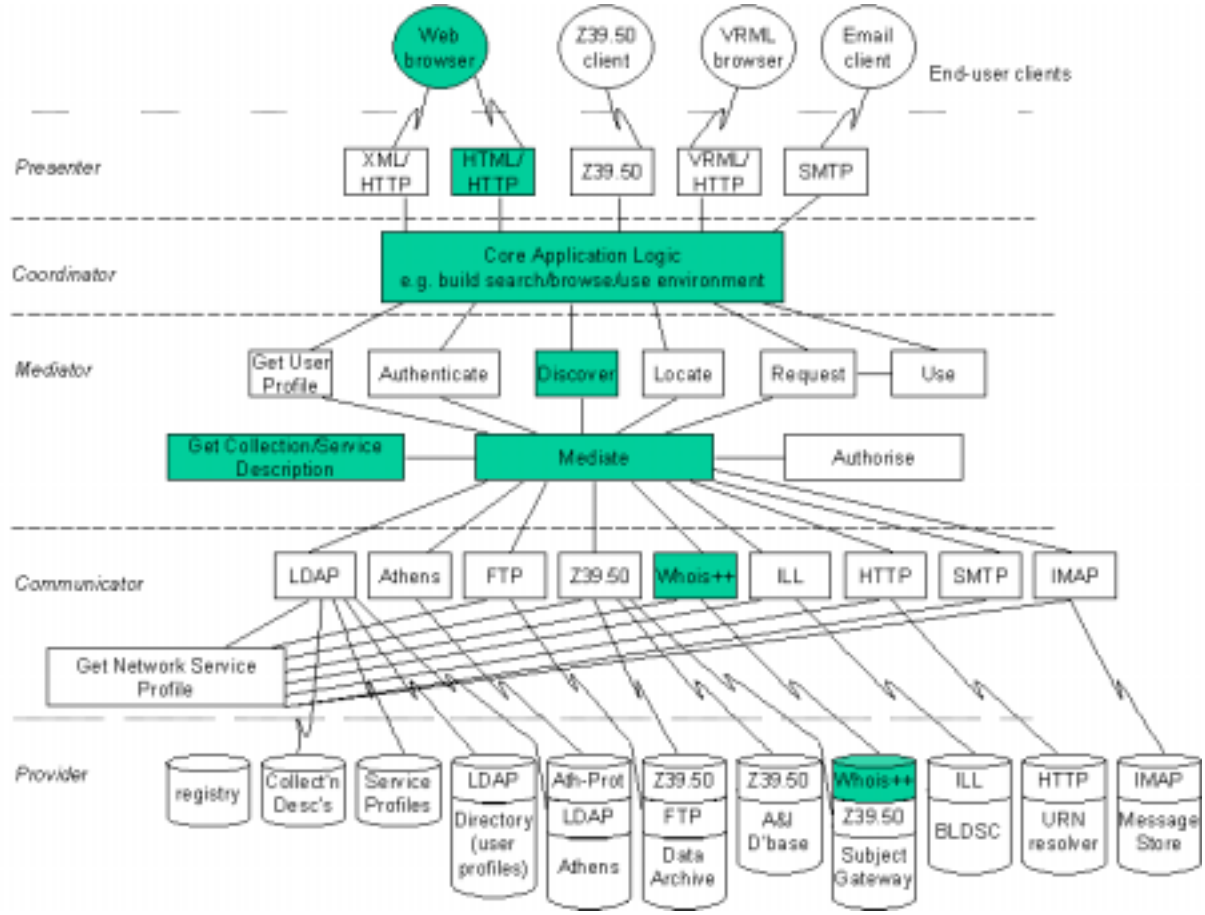


Figure 2.17: RDN ResourceFinder related to MIA

2.17.2 Logical architecture (MIA framework)

2.17.2.1 Presenter

The RDN ResourceFinder presents HTML/HTTP and WHOIS++ interfaces to the end-user. The HTML/HTTP interface is implemented as a CGI-based WHOIS++ client that communicates with the ResourceFinder WHOIS++ server.

2.17.2.2 Coordinator - Mediator - Communicator

The coordinator, mediator and communicator layers are responsible for taking a WHOIS++ query from the WHOIS++ component in the presenter layer and passing it in parallel to each of the WHOIS++ servers that the ResourceFinder knows about. Information about the available target WHOIS++ servers is held in ROADS configuration files. The mediator is responsible for merging and ranking the results obtained from each of the target WHOIS++ servers. If any of the target servers returns a 'too many results' error response, this error is returned through the ResourceFinder WHOIS++ server in the presenter layer - in this situation, any valid results from other target WHOIS++ servers are lost.

The mediator layer is responsible for gathering centroids from each of the target servers (on a nightly basis) and for comparing queries from the presenter layer against those centroids. Queries are only sent on to those WHOIS++ targets for which there appears to be a match in the centroid. Centroids are gathered using the WHOIS++ protocol.

2.17.2.3 Provider

The provider layer implements a WHOIS++ client. The mediator layer calls on this client to send WHOIS++ queries to each of the target WHOIS++ servers.

2.17.3 Technical architecture

2.17.3.1 Standards

The target WHOIS++ servers (SOSIG, Biz/ED, EEVL and OMNI) return records that are based on the ROADS DOCUMENT and SERVICE template types. In practice, on the Title, Description, Subject and URI attributes from these template types are displayed to the end-user.

2.17.3.2 Protocols

The ResourceFinder currently only supports the WHOIS++ protocol.

2.17.3.3 Software

The ResourceFinder is currently implemented using ROADS.

2.17.4 References

Resource Discovery Network: http://www.rdn.ac.uk/

Dempsey, L., 2000, The subject gateway: experiences and issues based on the emergence of the Resource Discovery Network. *Online Information Review*, 24 (1), 8-23.

Powell, A., 1999, *An MIA view of DNER portals*. Bath: UKOLN, the UK Office for Library and Information Networking. http://www.rdn.ac.uk/pubications/mia/

2.18 ROADS

Martin Hamilton, LUT

2.18.1 Introduction

The ROADS toolkit was originally developed with funding from the UK Electronic Libraries Programme (eLib) to build upon earlier work done by the Department of Computer Science at Loughborough University on developing a resource discovery system (working title 'debsearch') for SOSIG, the UK's Social Sciences information gateway.

2.18.1.1 Responsible agency

The ROADS project was a partnership between the Department of Computer Science at Loughborough University, the Institute of Learning and Research Technology (ILRT) at the University of Bristol and the UK Office for Library and Information Networking (UKOLN) at the University of Bath. However, the majority of actual software development and support was done at Loughborough.

2.18.1.2 Description/Scope

The ROADS project had two main goals:

- To provide a platform upon which the UK's subject gateways that were cataloguing Internet resources (originally ADAM, IHR-Info (History), OMNI, SOSIG and EEVL) could base their services.

- To research into the resource discovery issues which arose along the way.

The gateways were also funded via eLib, and the ROADS project was chartered to provide them with installation and technical support for the ROADS software as a core activity.

Over time the ROADS toolkit has been used in a number of other research projects and (pilot) services, including the European Commission-funded DESIRE project, a number of Trans-European Research and Education Networking (TERENA) projects, and several other countries' digital library research programmes. Central funding for ROADS ceased at the end of 1999, but the software continues to be developed and supported by its user community - following the model successfully used by the likes of the Linux operating system, the Apache Web server, and the Perl programming language.

The ROADS software tries not to impose a particular architecture upon its users. Instead, it provides a set of tools that can be used in any way that an operator of a ROADS based service finds convenient or useful. Kirriemuir, et al. (1998) describe how centroids and CIP may be used with ROADS to build a distributed resource discovery system.

Whilst ROADS provides a large number of tools to aid in the maintenance of ROADS databases, the ROADS user can easily import their data from another system (e.g. selected columns from a table in an SQL server), or use ROADS only to manage their metadata, and then export it to another system for presentation to the end user. Although ROADS offers a built-in search engine, there is no requirement that this be offered to a service's users. The predominant user interface to ROADS is based on HTML and HTTP, but there are also command line ('batch mode') tools.

Although ROADS has been presented as a modular toolkit, there are a number of inter-dependencies between modules and the programs which call them that prevent the ROADS modules from simply being reused in other packages. This is being addressed in an ongoing effort to properly modularise the ROADS codebase, but it currently poses a serious problem to anyone seeking to develop the ROADS toolkit to (for example) add support for new protocols and metadata formats.

2.18.1.3 Architectural diagram

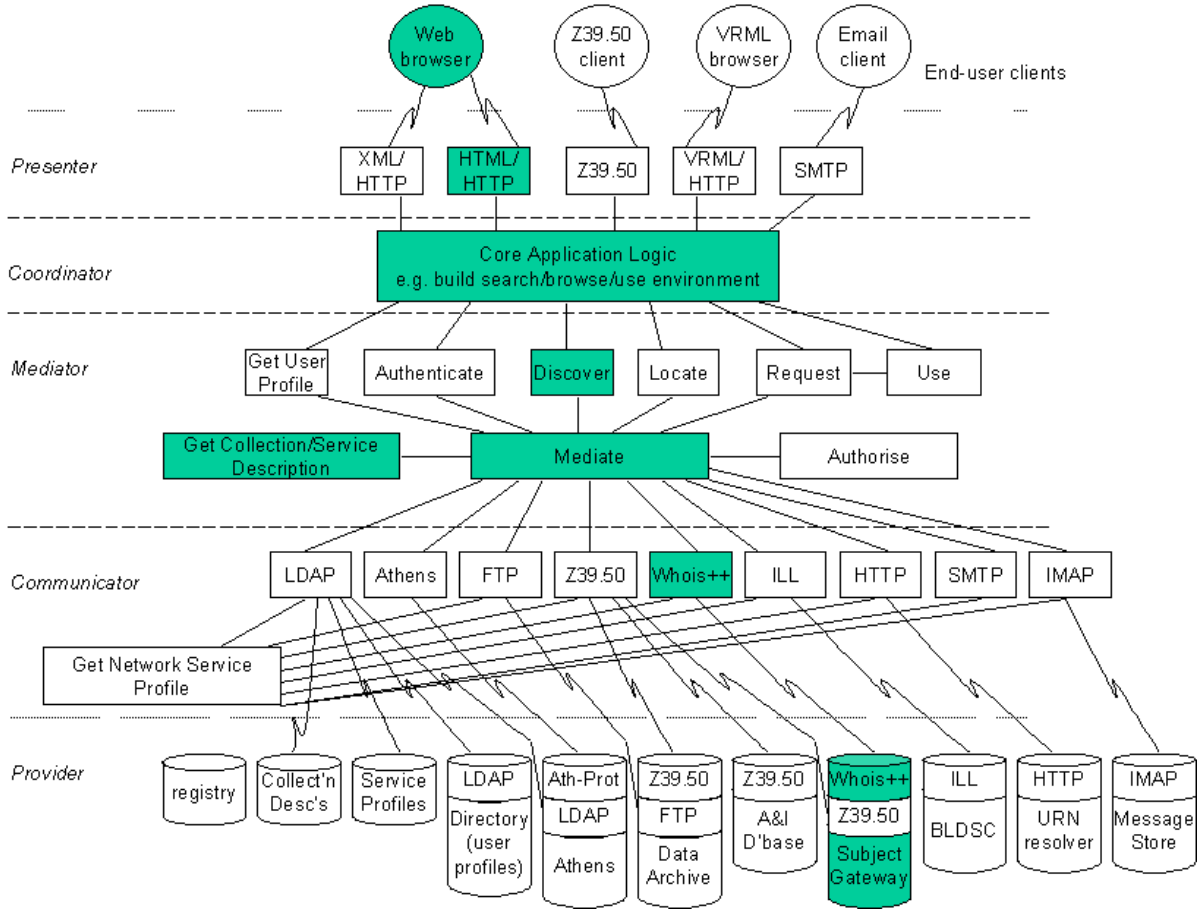


Figure 2.18: ROADS related to MIA

2.18.2 Logical architecture (MIA framework)

2.18.2.1 Presenter - Coordinator - Mediator - Communicator

CGI program cgi-bin/search.pl interacts with the end user and accesses the database also browsing via static HTML periodically regenerated and admin tools directly.

2.18.2.2 Provider

Can be ROADS WHOIS++ server bin/wppd.pl or any one of a number of other WHOIS++ servers. Must speak WHOIS++, though.

2.18.3 Technical architecture

2.18.3.1 Standards

ROADS is characterised by its support for the IETF's Common Indexing Protocol (CIP) and the WHOIS++ search and retrieval protocol - including centroids, its distributed indexing and searching mechanism (which pre-dated the Common Indexing Protocol).

In addition to the built-in WHOIS++ server (wppd), ROADS also comes with gateway software to add WHOIS++ search and centroid generation capabilities to other packages, including Harvest Brokers and Z39.50 servers (harvest_shim.pl and z3950_shim.pl) - code contributed by the CHIC-Pilot project. As in the CHIC-Pilot

case, the search and retrieval protocol and the metadata format are constrained, this time to WHOIS++ and IAFA templates.

Numerous programs exist to convert ROADS databases to and from a variety of formats (e.g. SOIF, GRS-1, LDIF, RDF and SQL), and to make different data sources available for searching via WHOIS++ (e.g. Harvest Brokers and Z39.50 servers).

2.18.3.2 Protocols

The core protocols used with ROADS are WHOIS++ (RFC 1835), centroids (RFC 1913 and RFC 1914), and HTTP for the user interface.

The Common Indexing Protocol, Harvest Gatherer/Broker and Z39.50 support (cross-searching and centroid generation) has not been used seriously to date and will likely require additional development/customisation for each usage scenario - e.g. to cope with the variations in Z39.50 profiles between installations.

2.18.3.3 Software

The libwww-perl CPAN module is needed for several of the ROADS tools, e.g. the Web based link checker. Harvest is required for the Harvest add-ons, and the zbatch program from the CNIDR Isite distribution is used for the Z39.50 gateway.

Additional databases may be plugged into ROADS using the WHOIS++ Gateway Interface (an invention of the ROADS developers which isn't supported by other WHOIS++ servers), though the only known databases which had this implemented were test servers.

The ROADS software can be downloaded from the Web site: http://roads.opensource.ac.uk.

It is open source software distributed under the terms of the GNU General Public License/Perl Artistic License (i.e. the standard Perl Terms and Conditions).

- Support: Community support is available via the open-roads@net.lboro.ac.uk mailing list. Commercial support and bespoke development may be available subject to negotiation - contact: roads-liaison@bristol.ac.uk

- Platforms: Any modern Unix-like system. Development and testing primarily done using Linux. Shrink-wrapped ROADS distribution available for exceptionally easy installation on Linux systems.

- Prerequisites: Perl, HTTP server optional for WWW interface.

2.18.4 References

ROADS software/documentation Web site: http://roads.opensource.ac.uk

ROADS project Web site: http://www.ilrt.bris.ac.uk/roads/

Kirriemuir, J., Brickley, D., Welsh, S., Knight, J., Hamilton, M., 1998, Cross-searching subject gateways: the query routing and forward knowledge approach. *D-Lib Magazine*, January. http://www.dlib.org/dlib/january98/01kirriemuir.html

2.19 UNIverse

Matthew J. Dovey, LAS

2.19.1 Introduction

2.19.1.1 Responsible agency

The UNIverse Consortium (17+ partners) - led by Fretwell-Downing Informatics (FDI). UNIverse was a project supported by the European Commission under the Telematics for Libraries Fourth Framework Programme (1996-1999). Its full title was "Large Scale Demonstrators for Global, Open Distributed Library Systems."

2.19.1.2 Description/Scope

UNIverse aims to provide services for a distributed virtual union library service, although its services are currently limited to search, retrieval and ordering services. It does provide additional features to aid searching and retrieval including support for the Z39.50 Explain Service, record format translation and thesauri facilities. It uses Z39.50 as it communications protocol making use of additional and extended services as required.

2.19.1.3 Architectural diagram

The basic UNIverse architecture is described as below:

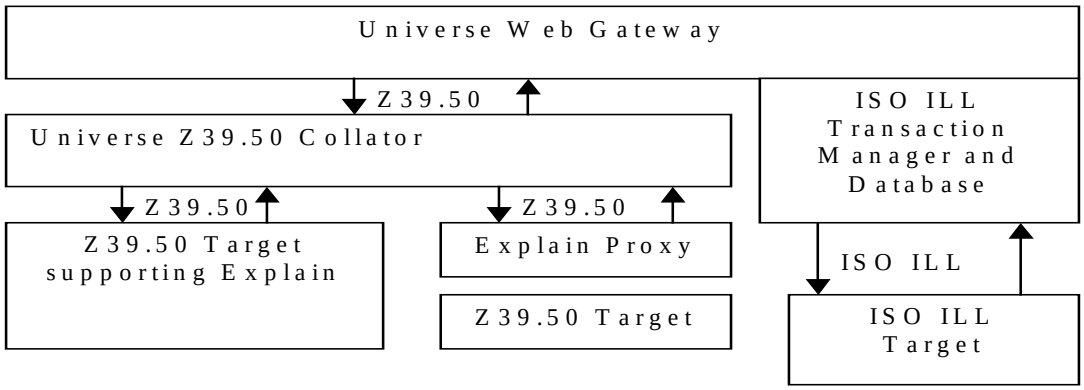


Figure 2.19: UNIverse architecture

2.19.2 Logical architecture (MIA framework)

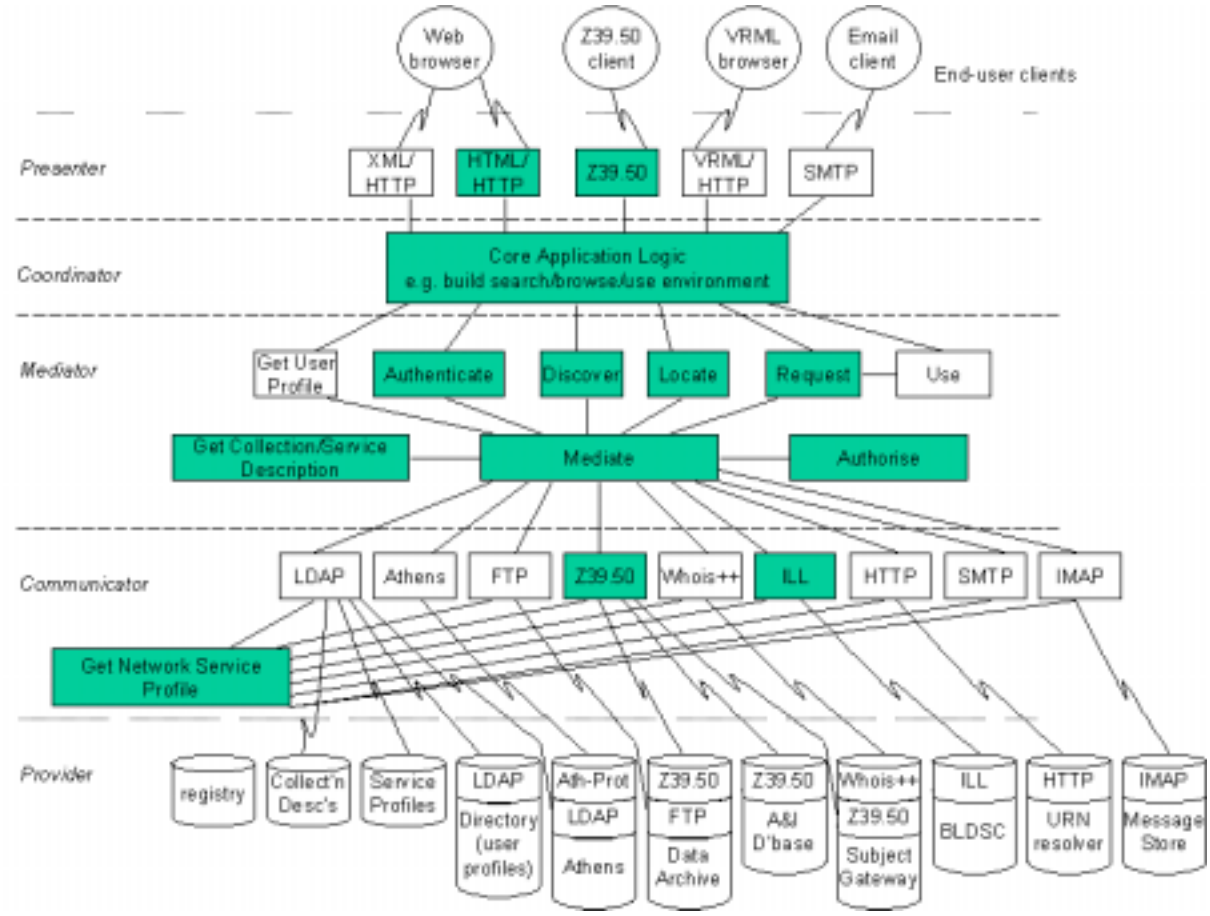


Figure 2.20: UNIverse related to MIA

2.19.2.1 Presenter

The presentation of the user interface to the end user is via HTTP and co-ordinated by the Web Gateway.

2.19.2.2 Coordinator

The bulk of the role of coordinator is also included within the functionality of the Web gateway software, although some of this in terms of delivery this is handled by the ISO ILL Manager/Database.

2.19.2.3 Mediator

Figure 2.20: UNIverse related to MIA

Some mediator functions are handled via the Web gateway, for example authentication. Functions related to searching and retrieval are handled via the Z39.50 collator. Functions relating to document order are handled via the ISO ILL Transaction Manager/Database. The Z39.50 collator also includes options for perform thesauri translations and record syntax translations. The Get Collection/Service Description function is provided via the Z39.50 Explain mechanisms.

2.19.2.4 Communicator

The communicator layers consist of the protocols Z39.50 for searching and retrieval and ISO ILL for item ordering. The Z39.50 Explain mechanism provides the Get Network Service Profile function.

2.19.2.5 Provider

It is expected that a provider will provide Z39.50 access, and also ISO ILL access if the ordering of items is supported. Z39.50 Explain is also needed for determining both collection/service information and network service information, however UNIverse includes an Explain proxy to provide this functionality to Z39.50 targets which do not support Z39.50 Explain.

2.19.3 Technical architecture

2.19.3.1 Standards

UNIverse is compliant with GEDI (Group on Electronic Data Interchange) and uses the ISO ILL standard and the Z39.50 standard including Explain.

2.19.3.2 Protocols

The HTTP, ISO ILL and Z39.50 protocols are used.

2.19.3.3 Software

The software is built on Index Data YAZ toolkit and also upon products from the consortium partners. Database facilities are provided via Oracle.

2.19.4 References

UNIverse Project: http://www.fdgroup.co.uk/projects/universe/

Kelly, A., 1998, *Project UNIverse: Large scale demonstrator for global, open distributed library service*. Paper given at: Convergence in the Digital Age: Challenges for Libraries, Museums and Archives, Amsterdam, 13-14 August 1998. http://www.cordis.lu/libraries/en/ifla/session1.html

Murray, R., Pettman, I., 1997, The UNIverse project. *New Library World*, 98 (2), pp. 53-59; also published in: *OCLC Systems and Services*, 13 (3), pp. 91-97.

3 CONSIDERATIONS TOWARDS DETERMINING AN ARCHITECTURAL MODEL

Matthew J. Dovey, LAS

3.1 Introduction

This paper outlines some architectural issues that Renardus will need to address at a fairly early stage in its development. It does not however propose any particular models or systems. It is assumed that the underlying goal is to provide a single point of access to a number of similar databases (in this case subject-based databases of internet resources). The term "gateway" here refers to such a system designed to provide such access to multiple databases.

3.2 Gateway

The gateway can be viewed as two distinct conceptual layers (although this may be implemented as a single system in certain cases). The first layer, "user interface" would deal with the generation of html forms and the communications to and from the users browser software. This layer would communicate with a second layer the "search engine" which would perform the searches, either via a distributed search mechanism or by querying a physical aggregate database.

3.3 Search Engine

There are two overall models which the Renardus subject gateway could adopt.

1. The first model is the physical union model. In this model Renardus would maintain its own database (or databases), which uses its own schema (possibly based on an appropriate standard). Participating data-sources (local gateways) would periodically export data from their own local databases, performing any format conversion necessary.

There are three models of how this centralised database (or databases) could be maintained:

a. Local record creation and centralised exports – resources are catalogued onto the local gateway, and periodically (e.g. weekly) records are exported from the local database to the central database(s). This model is typified by the CURL COPAC union catalogue.

b. Centralised record creation and local imports – resources are catalogued into the central database(s) and are periodically (e.g. weekly) imported from the central database(s) to the local database. This model is typified by the OCLC CORC project. An advantage of this approach is it would prevent the same resource being catalogued multiple times and ensures adherence to centralised controlled vocabularies and cataloguing rules.

c. Distributed cataloguing – record updates are sent to both local and central databases simultaneously. Typically this might be done via a gateway, which would distribute a single catalogue update (using the for example the Z39.50 Item Update service) to multiple servers, performing on the fly conversion and if necessary caching updates if a server is unavailable. Very little work has been done in this model to date, but it does reduce the delay in new records being made available on both local and central databases, which is inherent in the export/import models.

These models are not mutually exclusive and a mixture or hybrid approach may be applicable. In the export and distributed cataloguing models there is an element of conversion if the records are supplied in different formats. Syntactically conversion between the formats is relatively straightforward, however semantic conversion for example ensuring that a consistent vocabulary is used for terms would need substantial work to create and maintain the required thesauri and semantic crosswalks. These two models also could deal with duplicate records for a single resource by merging the records into a richer single record. All these models can ensure a consistent indexing policy by the fact that there is only one database.

Since this database would have to be accessible throughout Europe, it is unlikely that a single database would be practical. International Internet links are often not reliable or are slow. Therefore the database would need to be mirrored, typically a mirror for each participating country. There are four models of how this might be achieved:

a. Master-slave 1 – Updates can only be performed on a single master database. Changes are replicated to the mirror databases. Updates to the entire system depend on the accessibility of the master database.

b. Master-slave 2 – Updates can be performed on any on the mirrors. Changes are sent to the master database, which replicates to the other mirrors. This model has a danger of update conflicts (if the record is changed on different copies at the same time) but updates can be performed if the master is not accessible.

c. Peer-to-peer – Updates can be performed on any database, and changes are replicated to the other databases. This model has a danger of update conflicts (if the record is changed on different copies at the same time) and could be network intensive but updates can be performed if the master is not accessible.

d. Distributed cataloguing – if this model is adopted changes can be sent to all the database mirrors at the same time (with delayed caching if a database is inaccessible).

The above assumes that the mirrors will be identical in content. There may be a requirement for mirrors to have localised content. All of the above models are capable of supporting this, although some are better than others in this regard.

2. The second model is a virtual union model whereby Renardus acts as a gateway for cross-searching the local subject gateway databases. There are a number of issues with this approach:

a. It is an open question how many databases can be reliably cross-searched. Informed opinion places the maximum number anywhere from between 10 and 100!

b. There are two main approaches to attempting to reduce the number of databases that need to be searched, both based on forward knowledge of the database contents. The first is via fixed (often manually created) collection level metadata describing the specialities of the database (such as subject specialism). The second is by automated means such as periodically obtaining a centroid (which is in essence list of index terms for the database). Both are fairly experimental and neither are effective where the databases have similar content or have a general content with no particular specialism bias.

c. If the databases have no particular bias, and the indexes are relatively small compared to the records to be retrieved, a variation on the distributed mode would be to have centralised index (maintained in a fairly similar fashion to the models described in the above section of physical models of the gateway) but with decentralised databases of records. The query would then be performed on the central index, but subsequent record retrieval would be distributed.

d. There may be scope for a multi-tier approach. An international gateway could cross-search a number of national gateways which in turn cross-searches the individual subject databases. It is an open question as to how many such tiers can be reliably supported.

e. Cross-searching would involve a degree of query adaptation so that the users query can be modified to be consistent with the individual indexing policies of the databases being searched. This may need automatic thesaurus style expansion or conversion of the search term. There would be substantial effort involved in building the initial thesauri needed although experimental techniques in intelligent systems may reduce the subsequent maintenance needed.

f. Cross searching would also need to cope with a variety of input formats. This is not as difficult to deal with as query adaptation, as syntactic manipulation of the record to produce consistent displays would suffice in most cases (displaying consistent terms is less important than ensuring that indexes use consistent terms).

g. The cross-search would have to deal with duplicate entries. The simplest approach would be to only display the first record retrieved for a resource. The most useful approach would be to display merge records, but this would mean that all records would have to be retrieved from the databases before any results could be displayed to the end user (whereas the first approach means that records can be displayed as soon as they are received).

The physical and virtual models are not mutually exclusive and a hybrid model could be adopted. For instance each country could have a physical database with copies of international accessible databases but use a virtual search for nationally accessible databases.

3.4 User interface

There are three broad approaches that could be taken in implementing the user interface/Web server component:

1. Single interface – there is a single server (or localised cluster). Clearly if this approach is taken in conjunction with a physical database model then there is no need to use mirrors of the database. In effect there is just one system to access for the gateway located at one location (although it may be running on multiple machines at this location).

2. Mirror interface – there is a single interface, but this is mirrored on a number of servers throughout Europe so that downtime or slow international links do not prevent use of the gateway. Use of modern redirection techniques could be used so that the user only needs to know a single URL for accessing the gateway but is automatically (and transparently) redirected to use a local gateway.

3. Local interfaces – although the database(s) accessed are the same (or similar) there are local variance between interfaces. For instance each country could have an interface in the local language. There would be an easy means of reaching gateways from each other, but the user should not be redirected to a local gateway automatically (they may be using a non-local gateway for legitimate reasons). There need not be a 1-1 relationship with the search engines though. For instance there could be only one search engine (either a distributed or physical database model) located in one country, but localised interfaces (e.g. in different languages) in each country communicating with this one search engine.

4 CONCLUSIONS

Michael Day, UKOLN

4.1 Introduction

This report has reviewed eighteen broker models that have been designed to integrate access to heterogeneous and distributed information resources. It has attempted to use the MIA to provide a consistent view of the broker models reviewed. Some of the broker models (e.g. Agora) could be understood quite easily in MIA terms, others were harder to interpret because the publicly available information was not detailed enough or the assumptions behind the models were different. In any case, it can be difficult (at times) to understand the design assumptions made during the development of an architectural model from a finished product, e.g. a broker service. It was not clear in all these reviews where all of the component parts fitted into the five layers identified as part of the MIA.

4.2 Classification of the brokers reviewed

These caveats aside, it is possible to broadly divide the different models reviewed into the following four broad categories - on a continuum from simple to complex:

- The broker models that underlie open source indexing software toolkits like ASF Freeware, Harvest, ht://Dig, jake and ROADS.
- The broker models that underlie the cross searching of distributed Internet information gateways like the Finnish Virtual Library and the Resource Discovery Network (RDN) ResourceFinder. These currently tend to be based on open-source software like the ROADS toolkit and use relatively simple Internet protocols like WHOIS++ or LDAP.
- Broker models developed to handle more complex requirements, typically where more than one protocol and data format is in use. Some of those reviewed were based on - where possible - open source software, e.g. that developed for the EULER project. Some of the other systems are based to some extent on proprietary software and some have some dependence upon commercial products supplied by library software vendors. So, for example, the Agora Hybrid Library Management System (HLMS) is based on Fretwell-Downing Informatics's OLIB VDX system. CORC is based on proprietary software developed at OCLC, but could be licensed for use in a project like Renardus. These more complex models tend to be based on the action of standard protocols like Z39.50 and ISO ILL and sometimes need to interact with authentication services.
- Complex broker models being developed for information trading (GAIA).

4.3 Protocols

Outside of these broad categories, a tabular summary of the review findings (Table 4.1) makes it clear that a majority of the broker models reviewed used two protocols:

- HTTP - mostly within the Presenter layer to interact with a Web browser but also used within the Communicator and Provider layers of MIA
- Z39.50 - typically within the Communicator and Provider layers of MIA, occasionally within the Presenter layer.

Other protocols in use include ISO ILL (in Agora, DDB-NRW, GAIA and UNIverse), WHOIS++ (in the brokers based on the ROADS toolkit) LDAP (in Isaac Network and GAIA), NNTP, FTP and CIP.

	Presenter	**Communicator**	**Provider**
Agora	HTML/HTTP	Z39.50 ISO ILL	Z39.50 databases ISO ILL (BLDSC)

Aquarelle	HTML/HTTP	Z39.50	Z39.50 - Archive Database Z39.50 - Folder Server
ASF	HTTP	N/A	Z39.50 client
CHIC Pilot	HTML/HTTP	WHOIS++	WHOIS++ server
CORC	HTML/HTTP	HTTP	CORC database
DEF	HTML/HTTP	Z39.50	Z39.50 databases
DDB-NRW	HTML/HTTP	Z39.50 ISO ILL HTTP	Z39.50 library catalogues Z39.50 databases ILL - JASON/SUBITO HTTP collection database
ETB	XML/HTTP Z39.50	Z39.50 HTTP NNTP	Z39.50 databases HTTP - URN resolver NNTP metadata
EULER	HTML/HTTP Z39.50	Z39.50	Z39.50 - EULER service providers
FVL	HTML/HTTP	WHOIS++	WHOIS++ - subject gateway
GAIA	HTML Z39.50 E-mail etc.	Z39.50 FTP ISO ILL LDAP etc.	Underlying supplier services
Harvest	HTML/HTTP	Harvest Broker	Glimpse or other index backend
ht://Dig	HTML/HTTP	HTTP	HTTP - Web content
Isaac	HTML/HTTP	LDAP CIP	LDAP - metadata repository CIP - index service
jake	HTML/HTTP	N/A	MySQL or other SQL database
NCSTRL	Dienst	Dienst	Dienst
RDN ResourceFinder	HTML/HTTP	WHOIS++	WHOIS++ - subject gateway
ROADS	HTML/HTTP	WHOIS++	WHOIS++ - subject gateway

UNIverse	HTML/HTTP Z39.50	Z39.50 ISO ILL	N/A

Table 4.1 Summary of protocols used in broker models

4.4 Software

A variety of software products have been applied or developed to support the implementation of the broker models described. A summary of those identified can be found below (Table 4.2). Some of these are open-source (e.g. the ROADS toolkit or Combine), others are proprietary and would need to be licensed for use in Renardus.

	Software developed or used
Agora	Agora Hybrid Library Management System (based on FDI's VDX system)
Aquarelle	Aquarelle Resource Discovery System
ASF	ASFcrawl ASFserv ASFhttpd. CNIDR Isearch library (underlying search engine) YAZ (Index Data) Pavuk
CHIC Pilot	chic-search.pl
CORC	Mantis toolkit Kilroy Scorpion WordSmith
DEF	ZAP - a Z39.50 to Web gateway Z'mbol - a Z39.50 database system (Index Data and FDI) Combine
DDB-NRW	Digital library system developed by IHS Technologies and Axion Query Server (Dataware) WebPAC (Epixtech) BRS/Search (Dataware)

	MILOS II
ETB	Combine
EULER	HTTP-Z39.50 gateway - based on code from the EUROPAGATE project Z39.50 server and search system based on Zebra from Index Data
FVL	Uses the ROADS toolkit
GAIA	Developed in Java, built on CORBA
Harvest	Harvest Indexer
ht://Dig	htdig htsearch
Isaac	N/A
jake	jake
NCSTRL	Dienst
RDN ResourceFinder	Uses the ROADS toolkit
ROADS	ROADS toolkit
UNIverse	Built on Index Data YAZ toolkit Database facilities provided via Oracle

Table 4.2 Summary of software developed or used

PART IV - REMAINDER

5 REFERENCES

Baldonado, M., Chang, C.-C.K., Gravano, L., Paepke, A., 1997, The Stanford Digital Library metadata architecture. *International Journal on Digital Libraries*, 1 (2), 108-121.

Beagrie, N., 1999, *Convergence and integration online: the Arts and Humanities Data Service gateway and catalogues*. Paper delivered at Museums and the Web 1999, New Orleans, La., 11-14 March. http://www.archimuse.com/mw99/papers/beagrie/beagrie.html

Berggren, M., Brümmer, A, 1999, *Design Considerations for the EULER project*. Lund: NetLab. http://www.emis.de/projects/EULER/Reports/pD31.html

Davis, J.R., Lagoze, C., 2000, NCSTRL: design and deployment of a globally distributed digital library. *Journal of the American Society for Information Science*, 51 (3), 273-280.

Dempsey, L., 1999, The library, the catalogue, the broker: brokering access to information in the hybrid library. *New Review of Information Networking*, 5, 3-25.

Dempsey, L., 2000, The subject gateway: experiences and issues based on the emergence of the Resource Discovery Network. *Online Information Review*, 24 (1), 8-23.

Dempsey, L., Gardner, T., Day, M., Werf, T. van der, 1999, International information gateway collaboration: report of the first IMesh Framework Workshop. *D-Lib Magazine*, 5 (12), December. http://www.dlib.org/dlib/december99/12dempsey.html

Dempsey, L., Russell, R., 1997, 'Clumps' - or distributed access to scholarly material. *Program*, 31 (3), 239-249.

Dempsey, L., Russell, R., Murray, R., 1998, The emergence of distributed library systems: a European perspective. *Journal of the American Society of Information Scientists*, 49 (10), 942-951.

Dempsey, L., Russell, R., Murray, R., 1999, A utopian place of criticism? Brokering access to network information. *Journal of Documentation*, 55 (1), 33-70.

Deutsch, P., Emtage, A. Koster, M., Stumpf, M., 1994, *Publishing Information on the Internet with Anonymous FTP*. http://info.webcrawler.com/mak/projects/iafa/iafa.txt

Die Digitale Bibliothek NRW, 1998, *Konzept*. Bielefeld: Bibliothek der Universität Bielefeld. http://www.ub.uni-bielefeld.de/digibib-nrw/konzept.htm

Dörr, M., Christophides, V., Fundulaki, I., 1997, The specialist seeks expert views: managing folders in the Aquarelle project. In: *Museums and the Web 97*. Pittsburg, Pa.: Archives & Museum Informatics, 261-270.

Gardner, T., Iannella, R., 2000, Architecture and software solutions. *Online Information Review*, 24 (1), 35-39.

Gardner, T., Miller, P., Russell, R., 1999a, *The MIA logical architecture*, v. 0.3. Bath: UKOLN, UK Office for Library and Information Networking. http://www.ukoln.ac.uk/dlis/models/requirements/arch/

Gardner, T., Miller, P., Russell, R., 1999b, *MIA functional model*, v. 0.3. Bath: UKOLN, UK Office for Library and Information Networking. http://www.ukoln.ac.uk/dlis/models/requirements/func/

Greenstein, D., Murray, R., 1997, Metadata and middleware: a systems architecture for cross domain discovery. In: Greenstein, D. and Miller, P., eds., *Discovering online resources across the humanities.* Bath: UKOLN, the UK Office for Library and Information Networking, on behalf of the Arts and Humanities Data Service, 56-62. http://ahds.ac.uk/public/metadata/discovery.html

Groos, M., Hardt, J., Nold, A., Pieper D., Seiffert, F., Summann, F., 1998, *Die Digitale Bibliothek NRW - Technishes Konzept*. Bielefeld: Bibliothek der Universität Bielefeld. http://www.ub.uni-bielefeld.de/digibib-nrw/techkon.htm

Haberman, M., Heidbrink, S., 1999, Die Digitale Bibliothek NRW - Chronologie, Projektverlauf und Technische Beschreibung. *B.I.T. Online*, 2/1999. http://www.b-i-t-online.de/archiv/1999-02/nachrich/haberm/artikel.htm

Hickey, T.B., 2000, CORC: a system for gateway creation. *Online Information Review*, 24 (1), 49-53.

IEC 61360-1:1995, *Standard data element types with associated classification scheme for electric components -- Part 1: Definitions -- Principles and methods*. Geneva: International Electrotechnical Commission.

ISO 2709:1996. *Information and documentation -- Format for Information Exchange*. Geneva: International Organization for Standardization.

ISO 2788:1986, *Documentation -- Guidelines for the establishment and development of monolingual thesauri*. Geneva: International Organization for Standardization.

ISO 5964:1985, *Documentation -- Guidelines for the establishment and development of multilingual thesauri*. Geneva: International Organization for Standardization.

ISO 8879:1986, *Information processing -- Text and office systems -- Standard Generalized Markup Language (SGML)*. Geneva: International Organization for Standardization.

ISO 10160:1997, *Information and documentation -- Open Systems Interconnection -- Interlibrary Loan Application Service Definition*. Geneva: International Organization for Standardization.

ISO 10161-1:1997, *Information and documentation -- Open Systems Interconnection -- Interlibrary Loan Application Protocol Specification -- Part 1: Protocol specification*. Geneva: International Organization for Standardization.

ISO 10161-2:1997, *Information and documentation -- Open Systems Interconnection -- Interlibrary Loan Application Protocol Specification -- Part 2: Protocol implementation conformance statement (PICS) proforma*. Geneva: International Organization for Standardization.

ISO 23950:1998, *Information and documentation -- Information retrieval (Z39.50) -- Application service definition and protocol specification*. Geneva: International Organization for Standardization.

Koch, T., 2000, Quality controlled subject gateways: definitions, typologies, empirical overview. *Online Information Review*, 24 (1), 24-34.

Lagoze, C., 2000, *The Cornell Digital Library Research Group: architectures and policies for distributed digital libraries*. Invited Paper for DLW17, Tsukuba, Japan, February 2000. http://www.cs.cornell.edu/lagoze/papers/DLW17/cdlrg.htm

Langer, T., Adametz, H., Fellien, A., 1999, *Feasibility study: MALVINE information brokerage platform*. http://www.malvine.com/

Lassila O., Swick, R.R., eds., 1999, *Resource Description Framework (RDF) model and syntax specification*. W3C Recommendation. World Wide Web Consortium, 22 February. http://www.w3.org/TR/PR-rdf-syntax/

Lincke, D.M., Schmid, B. 1998, Mediating electronic product catalogs. *Communications of the ACM*, 41 (7), 86-88.

Lukas, C., Roszkowski, M., 1999, *The Isaac Network: LDAP and Distributed Metadata for Resource Discovery*. Paper presented at the IEEE Metadata 1999 conference. http://computer.org/proceedings/meta/1999/papers/46/clukas.html

Michard A., ed., 1998, *Final report: IE-2005 Aquarelle: sharing cultural heritage through multimedia telematics*. Le Chesnay: INRIA. http://aqua.inria.fr/Aquarelle/Public/EN/final-report.html

Moen, W.E., 1998, Accessing distributed cultural heritage information. *Communications of the ACM*, April, 41 (4), 45-48.

Online Computer Library Center, 1999, OCLC CORC project. *OCLC Newsletter*, 239, May/June. http://www.oclc.org/oclc/new/n239/index.htm#feature

Paepcke, A., Chang, C.-C.K., García-Molina, H., Winograd, T., 1998, Interoperability for digital libraries worldwide. *Communications of the ACM*, 41 (4), April, 33-43.

Paepke, A., Baldonado, M.Q.W., Chang, C.-C.K., Cousins, S., García-Molina, H., 1999, Using distributed objects to build the Stanford Digital Library Infobus. *IEEE Computer*, 32 (2), 80-87.

Paepke, A., Cousins, S.B., García-Molina, H., Hassan, S.W., Ketchpel, S.P., Röscheisen, M., Winograd, T., 1996, Using distributed objects for digital library interoperability. *IEEE Computer*, 29 (5), 61-68.

Powell, A., 1999, *An MIA view of DNER portals*. Bath: UKOLN, the UK Office for Library and Information Networking. http://www.rdn.ac.uk/pubications/mia/

RFC 1357: Cohen, D., 1992, *A format for e-mailing bibliographic records*. http://www.ietf.org/rfc/rfc1357.txt

RFC 1807: Lasher, R., Cohen, D., 1995, *A format for bibliographic records*. http://www.ietf.org/rfc/rfc1807.txt

RFC 1835: Deutsch, P., Schoultz, R., Faltstrom, P., Weider, C., 1995, *Architecture of the WHOIS++ service*. http://www.ietf.org/rfc/rfc1835.txt

RFC 1913: Weider, C., Fullton, J., Spero, S., 1996, *Architecture of the Whois++ Index Service*. http://www.ietf.org/rfc/rfc1913.txt

RFC 1914: Faltstrom, P., Schoultz, R., Weider, C., 1996. *How to interact with a Whois++ Mesh*. http://www.ietf.org/rfc/rfc1914.txt

RFC 2651: Allen, J., Mealling, M., 1999, *The Architecture of the Common Indexing Protocol (CIP)*. http://www.ietf.org/rfc/rfc2651.txt

RFC 2652: Allen, J., Mealling, M., 1999, *MIME Object Definitions for the Common Indexing Protocol (CIP)*. http://www.ietf.org/rfc/rfc2652.txt

RFC 2653: Allen, J., Leach, P. and Hedberg, R., 1999, *CIP Transport Protocols*. http://www.ietf.org/rfc/rfc2653.txt

Roszkowski, M., Lukas, C., 1998, A Distributed Architecture for Resource Discovery Using Metadata. *D-Lib Magazine*, September. http://www.dlib.org/dlib/june98/scout/06roszkowski.html

Russell, R., 1997, UKOLN MODELS 4: evaluation of cross-domain resource discovery. In: Greenstein, D. and Miller, P., eds., *Discovering online resources across the humanities.* Bath: UKOLN, the UK Office for Library and Information Networking, on behalf of the Arts and Humanities Data Service, 18-21. http://ahds.ac.uk/public/metadata/discovery.html

Sprick, A., Tröger, B., Hoffmann, L., Hupfer, G., 1999, *Das Metadatenformat der Collect-Datenbank der Digitalen Bibliothek NRW*. Cologne: Hochschulbibliothekszentrum des Landes Nordrhein-Westfalen (HBZ). http://www.hbz-nrw.de/DigiBib/dokumente/allg/meta.html

Valkenburg, Peter, ed., 1998, *Standards in a distributed indexing architecture*, draft version 1. 24 February. http://www.terena.nl/projects/chic-pilot/standards_v1.html

Valkenburg , P., Beckett , D., Hamilton, M., Wilkinson , S., 1998, *Standards in the CHIC-Pilot Distributed Indexing Architecture*. TERENA Networking Conference '98, Dresden, 7 October. http://www.terena.nl/projects/chic-pilot/tnc/paper.html

Van de Sompel, H., Lagoze, C., 2000, The Santa Fe Convention of the Open Archives Initiative. *D-lib Magazine*, 6 (2) February. http://www.dlib.org/dlib/february00/vandesompel-oai/02vandesompel-oai.html

Vinoski, S., 1997, CORBA: integrating diverse applications within distributed heterogeneous environments. *IEEE Communications Magazine*, 14 (2), 46-55.